Selbst-organisierende Datenstrukturen bei abhängigen Zugriffen

Frank Schulz

29. September 1995

Diplomarbeit nach einem Thema von Prof. Dr. Günter Hotz
in der Technischen Fakultät, Fachbereich 14 – Informatik,
der Universität des Saarlandes, Saarbrücken

Hiermit versichere ich an Eides Statt, daß ich diese Arbeit nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Saarbrücken, im September 1995

Inhaltsverzeichnis

Einleitung

Das Abspeichern von Informationen und der Zugriff auf gespeicherte Daten ist eine fundamentale Aufgabe in der Informatik. Wenn die Daten hierbei durch einen Namen angesprochen werden, so spricht man vom *Wörterbuchproblem*. Ein typisches Beispiel sind Telefonbücher. Die Namen der Fernsprech-Teilnehmer (Schlüssel) dienen dazu, die zugehörigen Telefonnummern (Informationen) zu identifizieren.

Selbst-organisierende Datenstrukturen stellen eine Möglichkeit zur Behandlung des Wörterbuchproblems dar. Sie erlauben es, die Darstellung der Daten im Speicher an das Verhalten des Benutzers dynamisch anzupassen und dadurch die erwartete Suchzeit zu reduzieren.
Selbst-organisierende Datenstrukturen bestehen aus zwei Komponenten: einer „normalen“ Datenstruktur zur vergleichsorientierten Suche (lineare Liste oder binärer Baum) und einer Umordnungs-Heuristik. Die Heuristik ist eine Regel, die gewisse Änderungen an der Datenstruktur vornimmt in der Hoffnung, dadurch künftige Anfragen billiger zu machen. Selbstorganisierende Datenstrukturen entfalten ihren Nutzen in zwei Bereichen:

(1) Die Zugriffe auf die Daten sind unabhängig voneinander und identisch verteilt, aber die Zugriffsverteilung ist unbekannt. In diesem Fall versuchen selbst-organisierende Datenstrukturen, die zugrundeliegende Zugriffsverteilung in einem gewissen Sinne zu approximieren.

(2) Das Nachfrageverhalten ändert sich mit der Zeit. Wenn das Verhalten des Benutzers nicht zeitinvariant ist, können selbst-organisierende Datenstrukturen dazu dienen, die Darstellung der Informationen im Speicher stets an den aktuellen Bedürfnissen auszurichten.

Während die in (1) beschriebene Situation bereits recht ausgiebig untersucht worden ist, existieren für das Szenario aus (2) nur wenige Analysen.
Es wurde mehrfach vorgeschlagen, die Abhängigkeit der Zugriffe durch eine Markov-Kette zu beschreiben. Bei diesem Modell hängt die Wahrscheinlichkeitsverteilung für den nächsten Zugriff von dem direkt vorausgehenden Zugriff (und nur von diesem) ab. Wenn die Heuristik kein Gedächtnis benutzen darf, wird man nicht hoffen können, jede

Folge abhängiger Zugriffe gut zu unterstützen. Man kann stets Folgen angeben, die zu schlechter mittlerer Zugriffszeit führen. Andererseits gibt es Situationen, in denen die mittlere Zugriffszeit bei abhängigen Nachfragen sehr klein ist. Es sind dies die Fälle, bei denen das Phänomen der *Lokalität* auftritt: eine kleine Teilmenge aller Elemente wird für lange Zeit ausschließlich nachgefragt, bevor zu einer anderen kleinen Teilmenge gewechselt wird. Das Übergangsverhalten der Anfragefolge ist *träge*. In diesem Fall können auch Heuristiken ohne Gedächtnis zu guten erwarteten Zugriffszeiten führen.

In dieser Arbeit benutzen wir die Idee der Rückwärtsanalyse, um selbst-organisierende Datenstrukturen zu analysieren. Dies basiert auf dem Gedanken, daß wir Informationen über das aktuelle Aussehen der Datenstruktur erhalten, indem wir uns ihre Vergangenheit ansehen.

Gliederung und Resultate

Im *ersten Kapitel* werden bekannte Ergebnisse über lineare Listen bei unabhängigen und bei abhängigen Zugriffen zusammengestellt. Wir stellen die Analysemethode mit absorbierenden Markov-Ketten als eine alternative Berechnungsmethode bei abhängigen Zugriffen vor, und führen mehrere Beispiele zur Nach-Vorne-Schieben-Regel bei abhängigen Zugriffen explizit aus.

Im *zweiten Kapitel* werden die selbst-organisierenden Bäume nach Allen/Munro beschrieben und ihr Verhalten für abhängige Zugriffe untersucht. Dabei benutzen wir die im ersten Kapitel eingeführten absorbierenden Markov-Ketten zur Analyse.
Es stellt sich heraus, daß in gewissen Fällen die erwartete Suchzeit innerhalb eines konstanten Faktors der Entropie der Zugriffsfolge liegt. Da die Entropie eine Abschätzung für die optimale asymptotische Suchzeit darstellt, folgt auch, daß die erwartete Suchzeit innerhalb eines konstanten Faktors der optimalen Suchzeit liegt. Dieser Faktor ist $2 \ln 2 \approx 1.3863$ wie bei unabhängigen Zugriffen.

Im *dritten Kapitel* werden die k-Mal-Hintereinander-Strategien für lineare Listen dargestellt und die Analyse von Gonnet/Munro/Suwanda auf abhängige Zugriffsfolgen verallgemeinert. Wir definieren eine neue Markov-Kette, indem die Zustände der ursprünglichen Kette zu k-Tupeln zusammengefaßt werden (sog. expandierte Markov-Kette). Mit diesem neuen Ansatz folgen die bekannten Resultate bei unabhängigen Zugriffen als einfaches Korollar.
Bei der Analyse abhängiger Zugriffsfolgen mit trägem Übergangsverhalten wird die intuitive Vermutung bestätigt, daß die erwartete Suchzeit umso schlechter wird, je länger die Aktion der Heuristik verzögert wird. Weiter finden wir den Zusammenhang, daß die Hierarchie der Heuristiken bzgl. der erwarteten Suchzeiten gerade umgekehrt ist als bei unabhängigen Zugriffen.

Das *vierte Kapitel* behandelt die Analyse weiterer Regeln für lineare Listen bei abhängigen Zugriffen und den Vergleich der erwarteten Suchzeiten. Insbesondere wird auf die Vertauschen-Regel und die k-Plätze-Vorwärts-Regeln eingegangen.
Wir betrachten eine spezielle Klasse von Zugriffsfolgen, um den zugrundeliegenden Zustandsraum zu reduzieren. Dieser Berechnungsansatz wurde bei unabhängigen Zugriffen mehrfach gewählt. Wir wenden ihn auf abhängige Zugriffsfolgen an und können dadurch erstmals weitere Heuristiken ohne Gedächtnis neben der Nach-Vorne-Schieben-Regel analysieren. Wir finden, daß bei trägen Übergängen auch hier die Hierarchie der Heuristiken umgedreht ist, verglichen mit unabhängigen Zugriffen.

Der *Anhang A* gibt eine Zusammenfassung der benötigten Begriffe und Sätze aus der Theorie der Markov-Ketten.

An dieser Stelle möchte ich mich bei Herrn Professor Dr. Hotz für die Vergabe der Diplomarbeit und die interessante Themenstellung herzlich bedanken.

Mein besonderer Dank gilt Herrn Dr. Elmar Schömer für viele hilfreiche Diskussionen und die sorgfältige Durchsicht der Arbeit.

Meine Eltern haben durch Verständnis und Unterstützung wesentlich zum Gelingen dieser Arbeit beigetragen, ihnen gilt mein größter Dank.

Kapitel 1

Lineare Listen

1.1 Problemstellung

Wir betrachten das statische Wörterbuchproblem. Die gesamte abzuspeichernde Menge ist von Anfang an bekannt, es werden weder Elemente hinzugefügt noch entfernt. Die einzige Operation ist der Zugriff auf Elemente der Menge. Weiter setzen wir voraus, daß alle Suchvorgänge erfolgreich sind, d.h. daß nur Elemente nachgefragt werden, die sich auch tatsächlich in der gespeicherten Menge befinden. Alle Elemente sind verschieden.

Es genügt also, im folgenden nur noch die Menge

$$S = \{1, 2, \ldots, n\}$$

mit n Elementen zu betrachten. Diese stellen die Schlüssel dar, nach denen wir suchen, und repräsentieren z. B. die Werte der Schlüsselfelder von Datensätzen in einer Datenbank. Die damit verknüpfte Nutz-Information spielt bei der Suchstrategie keine Rolle und wird vernachlässigt. Die Daten sind in einer vorwärtsverketteten Liste gespeichert. Die Suche beginnt beim Listenkopf und schreitet solange fort, bis das gesuchte Element gefunden ist. In jedem Schritt wird das aktuell betrachtete Listenelement mit dem zu suchenden Wert verglichen. Die Kosten für die Suche sind gleich der Anzahl der benötigten Vergleiche (= Position des Elements in der Liste). Wir betrachten verschiedene Zugriffsfolgen und die dazugehörigen erwarteten Kosten für die Suche. Die einzelnen Nachfragen können unabhängig voneinander sein, oder voneinander abhängen, etwa so, daß durch das zuletzt gesuchte Element angegeben wird, wie die Wahrscheinlichkeitsverteilung für das nächste zu suchende Element aussieht.

1.2 Unabhängige Zugriffe

Es gebe eine Wahrscheinlichkeitsverteilung

$$\vec{p} = (p_1, \ldots, p_n)$$

auf S, so daß das Element i mit der Wahrscheinlichkeit p_i nachgefragt werde. Die Zugriffe sind unabhängig und identisch verteilt mit der Verteilung $\vec{p}$. Ohne Beschränkung der Allgemeinheit habe jedes Element eine positive Nachfragewahrscheinlichkeit. Wenn dies nicht so wäre, so würden bei allen betrachteten Verfahren nichtbenötigte Elemente im Laufe der Zeit ans Ende der Liste gesetzt und dort verbleiben. Sie spielten also bei der asymptotischen Betrachtung keine Rolle mehr.

1.2.1 Optimale Strategien

Wenn man die Wahrscheinlichkeitsverteilung $\vec{p}$ kennt, dann wird man die Liste so anordnen, daß das am häufigsten nachgefragte Element an erster Position steht und die Liste nach absteigenden Zugriffswahrscheinlichkeiten sortiert ist. Sei o.B.d.A. $p_1 \geq p_2 \geq \cdots \geq p_n$. Dann ist

$$M = \sum_{i=1}^{n} i \cdot p_i$$

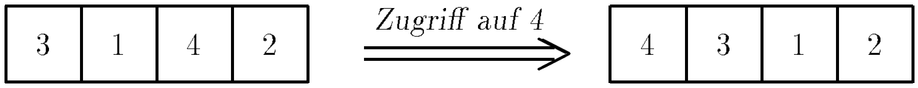
der Erwartungswert für die Kosten einer Suche bei optimaler statischer Ordnung der Liste.

Wenn die Wahrscheinlichkeitsverteilung $\vec{p}$ nicht bekannt ist, dann kann man mit einer beliebigen Ordnung der Liste starten und für jedes Element mitzählen, wie oft es nachgefragt wurde. Dann wird man die Liste nach absteigenden Zählerwerten sortieren und immer wieder anpassen, wenn ein Zähler einen anderen Zähler überrundet hat (*counter scheme*). Nach dem (starken) Gesetz der großen Zahlen konvergieren die relativen Häufigkeiten gegen die tatsächliche Wahrscheinlichkeitsverteilung.
Bei unbekannter Verteilung $\vec{p}$ ist dieses Verfahren optimal, es benötigt aber n Zähler (die potentiell beliebig groß werden können).

Die Vorgehensweise ist nicht nur asymptotisch optimal, sondern auch dann, wenn alle Strategien nur beschränkt viel Speicher benutzen dürfen (Lam/Siu/Yu [27]), oder wenn man nur einen endlichen Ausschnitt der Zugriffsfolge sieht (Hofri/Shachnai [19]).

1.2.2 Die Nach-Vorne-Schieben-Regel

Wenn man ohne diese Zähler auskommen möchte, so kann man folgende Umordnungs-Regel benutzen: immer wenn ein Element nachgefragt wurde, wird es von seiner aktuellen Position der Liste gelöscht und an den Anfang der Liste gesetzt. Wenn es sich schon am Listenkopf befindet, so geschieht nichts. Dadurch erreicht man, daß sich häufig benutzte Elemente meistens im vorderen Bereich der Liste aufhalten, während selten benötigte Werte rasch ans Ende der Liste abdriften.
In folgendem Beispiel soll der Listenkopf links sein, und die Liste von links nach rechts durchsucht werden.



Diese Regel wird *Nach-Vorne-Schieben-Regel* oder *move-to-front* (MTF) genannt. Sie wurde erstmals 1965 analysiert (McCabe [28]). Der Aufwand für die Umordnung ist proportional zum Suchaufwand, wenn man die Liste in einem Array speichert. Jedes Element, das während der Suche betrachtet wurde, wird bei der Umordnung bewegt. Wenn die Liste mit Zeigern realisiert ist, bleibt der Aufwand konstant.

1.2.2.1 Erwartete Suchkosten

Satz 1.1:

> *Die erwarteten Suchkosten bei der Nach-Vorne-Schieben-Regel sind höchstens doppelt so groß wie die erwarteten Suchkosten bei optimaler statischer Anordnung.*

Um diese Aussage zu zeigen, machen wir folgende Beobachtung:

Lemma 1.2 (Strukturlemma für die Nach-Vorne-Schieben-Regel)
Das Element j steht vor i in der Liste, wenn der letzte Zugriff auf j nach dem letzten Zugriff auf i erfolgt ist.

Beweis:
Durch einen Zugriff auf j wird dieses Element an den Anfang der Liste gesetzt, steht also vor i. Wenn danach kein Zugriff auf i mehr erfolgt, bleibt die relative Ordnung von i und j erhalten. ■

Mit Hilfe dieser Beobachtung zeigen wir Satz 1.1.

Beweis:
Seien für die Nach-Vorne-Schieben-Regel definiert

$b(j,i)$	:	asymptotische Wahrscheinlichkeit, daß j vor i steht
$\mu(i)$	:	mittlere Position von i = mittlere Suchzeit für die Suche nach i
μ	:	mittlere Suchzeit bei der Nach-Vorne-Schieben-Regel

Wir betrachten die Wahrscheinlichkeit des Ereignisses, daß j vor i steht. Gemäß dem Strukturlemma gibt es dann ein $k \geq 0$, so daß die letzten $k+1$ Zugriffe aus einem Zugriff auf j gefolgt von k Nachfragen für Elemente verschieden von i und j bestanden:

$$\begin{aligned} b(i,j) &= p_i \cdot \sum_{k \geq 0} (1-(p_i+p_j))^k \\ &= \frac{p_i}{p_i+p_j}. \end{aligned}$$

Die erwartete Position $\mu(i)$ des Elementes i ist gegeben durch

$$\begin{aligned} \mu(i) &= 1 + (\text{erwartete Anzahl von Elementen, die vor } i \text{ stehen}) \\ &= 1 + \sum_{j \neq i} b(j,i). \end{aligned}$$

Die durchschnittliche Suchzeit mit der MTF-Regel beträgt somit

$$\begin{aligned} \mu &= \sum_{i=1}^{n} p_i \cdot \mu(i) \\ &= \sum_{i=1}^{n} p_i \cdot \left(1 + \sum_{j \neq i} \frac{p_i}{p_i+p_j}\right) \\ &= 1 + \sum_{i=1}^{n} \sum_{j \neq i} \frac{p_i \cdot p_j}{p_i+p_j} \\ &= 1 + 2 \cdot \sum_{i<j} \frac{p_i \cdot p_j}{p_i+p_j}. \end{aligned} \tag{1.1}$$

Dieses Ergebnis kann man mit der optimalen statischen Ordnung vergleichen:

$$\begin{aligned} \mu &= 1 + 2 \cdot \sum_{i<j} p_j \cdot \frac{p_i}{p_i+p_j} \\ &\leq 1 + 2 \cdot \sum_{i<j} p_j \cdot 1 \\ &= 1 + 2 \cdot \sum_{j} p_j \cdot (j-1) \\ &= 1 + 2 \cdot (\sum_{j} p_j \cdot j) - 2 \\ &= 2 \cdot M - 1. \end{aligned} \tag{1.2}$$

■

Die obige Darstellung folgt Mehlhorn [29]. Die Formel (1.1) wurde erstmals von McCabe [28] angegeben. Die Idee der Benutzung von Indikator-Zufallsvariablen zur Vereinfachung der Berechnung und der Vergleich mit der optimalen Suchzeit (1.2) stammen von Burville/Kingman [8].

1.2.2.2 Zusammenhang mit Markov-Ketten

Der Analyse liegt die Theorie der Markov-Ketten zugrunde (siehe Anhang A). Die linearen Listen unter der Nach-Vorne-Schieben-Regel bilden eine ergodische Markov-Kette. Jede Permutation der Liste ist ein Zustand der Kette, der Zustandsraum ist also die Menge der Permutationen $\mathcal{S}_n$ von $\{1, \ldots, n\}$, im folgenden auch als Konfigurationen bezeichnet.
Einen Übergang von π nach σ gibt es, wenn ein Element i existiert, so daß aus der Konfiguration π die Konfiguration σ entsteht, indem man das Element i an den Anfang der Liste schiebt. Dieser Übergang hat die Wahrscheinlichkeit $p_{\pi\sigma} = p_i$.
Diese spezielle Markov-Kette ist ergodisch. Denn durch eine geeignete Folge von Zugriffen kann man jede Permutation der Liste aus jeder Start-Permutation erzeugen ($\Longrightarrow$ Kette ist irreduzibel), und wenn man eine Permutation erreicht hat, kann man in dieser beliebig lange bleiben durch Zugriffe auf das erste Element der Liste ($\Longrightarrow$ Kette ist azyklisch). Dies gilt, da nach Voraussetzung alle Elemente i eine Nachfragewahrscheinlichkeit $p_i > 0$ besitzen.
Die Ergodizität der Kette ist der Grund für die Existenz der $b(j, i)$. Denn da die stationären Wahrscheinlichkeiten $q(\pi)$ existieren, sind auch die

$$b(j, i) = \sum_{\{\pi : \pi(j) < \pi(i)\}} q(\pi)$$

wohldefiniert.
Man kann die stationäre Wahrscheinlichkeit $q(\pi)$ für das Ereignis, sich in Konfiguration π der Liste zu befinden, auch explizit ausrechnen (Hendricks [17, 18], siehe auch Phatarfod [31, 32]). Für die folgenden Überlegungen benötigen wir dies aber nicht.

Durch die Regeln zur Selbst-Organisation werden „gute“ Konfigurationen (solche mit geringer erwarteter Suchzeit) mit hoher Wahrscheinlichkeit auftreten. Man kann aber nicht folgern, daß nach einer gewissen Zeit *jede* Konfiguration gut ist. Die Frage, wie man mit Selbst-Organisations-Regeln zu einer guten statischen Konfiguration kommt, wurde von Richards [34] behandelt.

1.2.2.3 Weitere Ergebnisse

Das Ergebnis von Satz 1.1 ist unabhängig von der speziellen Verteilung $\vec{p}$. Wenn man die Wahrscheinlichkeitsverteilung $\vec{p}$ kennt, so kann man diese Schranke noch verschärfen.

Sei $H_n = \sum_{i=1}^{n} 1/i$ die n-te harmonische Zahl (es gilt: $\ln n \leq H_n \leq 1 + \ln n$) und $H_n^{(l)} = \sum_{i=1}^{n} i^{-l}$ die n-te harmonische Zahl l-ter Ordnung.

Satz 1.3 (Gonnet/Munro/Suwanda 1981)

1. *Verteilung nach Zipf*
   Die Zugriffswahrscheinlichkeit für das Element i ist $p_i = 1/(iH_n)$. Es gilt $M = n/H_n$ und
   $$\mu \leq 2 \ln 2 \cdot M.$$
2. *Exponentielle Verteilung*
   Die Zugriffswahrscheinlichkeit für das Element i ist $p_i = (1-a)a^{i-1}$ für ein a, $0 \leq a \leq 1$. Für den Grenzwert bei $n \to \infty$ gilt
   $$\mu \leq 2 \ln 2 \cdot M.$$
3. *Verteilung nach Lotka*
   Die Zugriffswahrscheinlichkeit für das Element i ist $p_i = 1/(i^2 H_n^{(2)})$. Dann ist $M = H_n/H_n^{(2)}$ und
   $$\mu \leq \pi/2 \cdot M.$$
   Außerdem gilt
   $$\mu/M \to \pi/2 \text{ bei } n \to \infty.$$

■

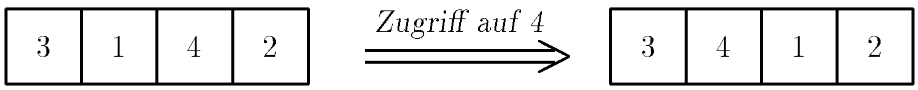
Diese Aussagen werden in [16] gezeigt. Die Verteilung nach Lotka ist ein Beispiel für das schlimmste, was passieren kann. Denn es gilt folgendes Ergebnis, das den Satz 1.1 verbessert [12].

Satz 1.4 (Chung/Hajela/Seymour 1985)
Für alle Folgen von unabhängigen identisch verteilten Zugriffen auf die Listenelemente sind die erwarteten Suchkosten bei der Nach-Vorne-Schieben-Regel höchstens um den Faktor $\pi/2$ größer als die minimalen erwarteten Suchkosten, und diese Schranke ist scharf.

■

1.2.3 Die Vertauschen-Regel

Eine weitere bekannte Umordnungsregel neben der MTF-Heuristik ist die Vertauschen-Regel oder *transposition* (TR). Hier wird das gerade nachgefragte Element nicht an den Anfang der Liste gesetzt, sondern nur um einen Platz in Richtung des Listenkopfs bewegt. Wenn es sich schon dort befindet, so passiert nichts.



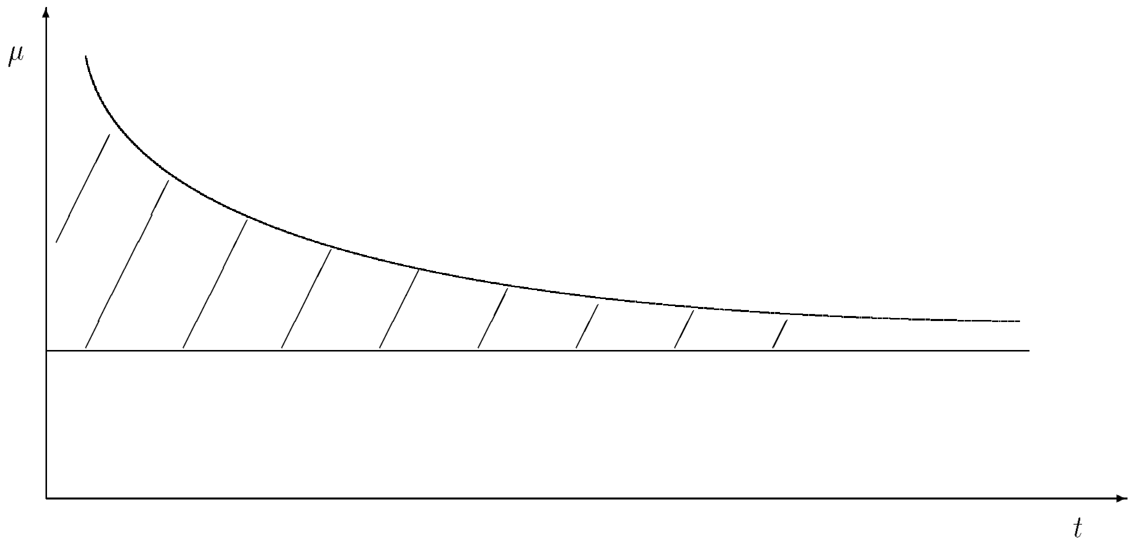
Diese Regel eignet sich besser für den Einsatz in Feldern (Arrays), denn man vertauscht nur zwei benachbarte Elemente miteinander. Der Aufwand für die Umordnung ist also konstant. Bei der Nach-Vorne-Schieben-Regel ist er aber proportional zur Suchzeit: alle Elemente zwischen dem ersten und dem gerade nachgefragten Element müssen um einen Platz nach hinten versetzt werden.
Es gilt folgendes Ergebnis aus [35].

Satz 1.5 (Rivest 1976)
Für $n > 2$ und alle Verteilungen $\vec{p}$ außer der Gleichverteilung hat die Vertauschen-Regel geringere erwartete Suchkosten als die Nach-Vorne-Schieben-Regel.

■

1.2.4 Konvergenzgeschwindigkeit

Ein weiteres Kriterium für die Beurteilung der Heuristiken neben der asymptotischen erwarteten Suchzeit ist die Konvergenzgeschwindigkeit. Wie lange braucht die Regel, um aus einer beliebigen Anfangspermutation der Liste das stationäre Gleichgewicht zu erreichen?
Der Begriff der *Mehrarbeit* (*overwork*) wurde von Bitner [7] eingeführt. Darunter versteht man den Aufwand, der über dem asymptotischen Aufwand hinaus in der Anfangsphase betrieben werden muß. In folgendem Diagramm ist die Mehrarbeit als schraffierte Fläche dargestellt.



Unter der Annahme, daß am Anfang jede Ordnung der Liste gleichwahrscheinlich ist, kann man eine Formel für den erwarteten Suchwaufwand nach t Schritten herleiten. Für $t \to \infty$ konvergiert er gegen den asymptotischen Suchaufwand. Die Differenz, summiert über alle t, ergibt gerade die Mehrarbeit.
In [7] wurde keine allgemeine Aussage gezeigt. Für alle betrachteten Verteilungen stellte sich aber heraus, daß die Mehrarbeit bei der Vertauschen-Regel um einen Faktor n größer ist als bei der Nach-Vorne-Schieben-Regel.

Wir beschäftigen uns im folgenden nur noch mit der asymptotischen erwarteten Suchzeit.

1.2.5 Weitere Regeln

In der Literatur werden zahlreiche weitere Regeln angegeben. Vier Beispiele sind die folgenden Klassen von Heuristiken.

1.2.5.1 Move-Up(k)

Die k-Plätze-Vorwärts-Regeln oder *move-up(k)*-Regeln sind zwischen der Nach-Vorne-Schieben-Regel und der Vertauschen-Regel angesiedelt. Das nachgefragte Element wird um k Plätze nach vorne bewegt. Wird dabei der Anfang der Liste erreicht, so wird das Element dort plaziert. Dadurch wird gewissermaßen ein Spektrum von Regeln definiert, dessen Endpunkte die Vertauschen-Regel ($k = 1$) und die Nach-Vorne-Schieben-Regel ($k = n - 1$) bilden.
Diese Regeln wurden erstmals von Rivest [35] betrachtet.

1.2.5.2 Pos(k)

Bei der pos(k)-Regel wird ein nachgefragtes Element auf Platz j um eine Position weiter nach vorne gerückt für $j \leq k$, und auf Platz k gesetzt, wenn es sich auf $j > k$ befand. Die Liste wird also in zwei Bereiche $1, \ldots, k$ und $k, \ldots, n$ unterteilt. Auf dem ersten Bereich wird die Vertauschen-Regel angewendet, auf dem zweiten Bereich die Nach-Vorne-Schieben-Regel. Der Platz k der Liste stellt den Übergangsbereich dar.
Eine Begründung für diese Regeln ist der Versuch, die gute asymptotische Suchzeit der Vertauschen-Regel mit der guten Konvergenzgeschwindigkeit der Nach-Vorne-Schieben-Regel zu verbinden.

1.2.5.3 Switch(k)

Diese Regeln sind symmetrisch zu den pos(k)-Regeln aufgebaut. Ein nachgefragtes Element auf Platz j wird für $j \leq k$ an den Anfang der Liste gesetzt, und für $j > k$ um

eine Position nach vorne bewegt. Also wird auf dem ersten Bereich der Liste die Nach-Vorne-Schieben-Regel und auf dem zweiten Bereich die Vertauschen-Regel benutzt.
Die SWITCH(k)-Regeln kann man intuitiv dadurch begründen, daß ein nachgefragtes Element aus dem hinteren Bereich sich erst „bewähren“ soll, bevor es an den Anfang der Liste gesetzt wird und damit evtl. Schaden anrichtet.

Diese Regeln wurden von Arnaud [4] eingeführt und von Tenenbaum/Nemes [36] und Lam [25] für eine spezielle Zugriffsverteilung analysiert (siehe Satz 1.7).

1.2.5.4 DOUBLE AREA(a, k)

Diese Klasse von Regeln teilen die Liste in zwei Bereiche der Größen a und $n - a$ ein für einen festen Parameter a. Im ersten Bereich (*area of control*) wird die Nach-Vorne-Schieben-Regel angewendet. Dieser Bereich ist nochmals unterteilt durch den Parameter $k < a$ in die erste Zone (*primary area*) und die Test-Zone (*trial area*). Ein Element, das erst im zweiten Bereich gefunden wurde, wird an den Anfang der Test-Zone gesetzt, auf Position k des ersten Bereichs. Dort muß es sich bewähren, bevor es in die erste Zone gelangt.

Diese Regeln sowie empirische Ergebnisse wurden von Ch'ng/Srinivasan/Ooi [11] angegeben.

1.2.5.5 Darstellung durch Permutationen

Es ist sinnvoll, eine einheitliche Sichtweise der Regeln zu benutzen.
Eine Heuristik ohne Gedächtnis trifft ihre Entscheidung nur aufgrund der Position, auf der das gesuchte Element gefunden wurde. Aufgrund dieser Information wird die Liste irgendwie umgeordnet, und zwar ohne Ansehen der Elemente. Die Wirkung der Umordnung kann man auffassen als Anwendung einer Permutation auf die bisherige Konfiguration der Listenelemente. Da das gesuchte Element auf n verschiedenen Positionen sein kann, sind n verschiedene Permutationen möglich.
Eine Heuristik kann also beschrieben werden durch eine Folge von n Permutationen $\{\tau_1, \ldots, \tau_n\}$. Eine Liste in Konfiguration π wird umgeordnet zu $\pi \circ \tau_j$, wenn auf das Element auf Position j der Liste zugegriffen wurde.
Oder anders ausgedrückt: wenn auf Element i zugegriffen wird, befindet es sich gerade an Position $j = \pi^{-1}(i)$. Die Liste wird dann umsortiert zu $\pi \circ \tau_{\pi^{-1}(i)}$.
Mit diesem Verständnis kann man beispielsweise die Nach-Vorne-Schieben-Regel beschreiben (siehe auch [26]) als:

$$\tau_j(k) = \left\{ \begin{array}{ccl} j & : & \text{falls } k = 1 \\ k-1 & : & \text{falls } 1 < k \leq j \\ k & : & \text{falls } j < k \leq n \end{array} \right\}$$

1.2.6 Vergleich von Regeln

Mit der Definition verschiedener Regeln taucht sofort die Frage nach dem Vergleich zweier Regeln auf. Außerdem kann man sich fragen, welche Regel aus einer gewissen Menge von Heuristiken optimal ist, oder ob überhaupt eine optimale Regel existiert. Dabei kann es sinnvoll sein, dem Vergleich nur eine kleine Menge von Wahrscheinlichkeitsverteilungen zugrundezulegen.

Im Artikel [35] hat Rivest vermutet, daß die k-Plätze-Vorwärts-Regeln für sinkende k fallende erwartete Suchkosten aufweisen, und daß die Vertauschen-Regel minimale erwartete Suchkosten unter allen Heuristiken ohne Gedächtnis besitzt.

Diese Vermutungen sind in ihrer Allgemeinheit unbewiesen. Es gibt aber folgende Teilergebnisse.

Gegenbeispiel 1.6 (Anderson/Nash/Weber 1982)

> *Es gibt eine Selbst-Organisation-Regel ohne Gedächtnis für lineare Listen, die bei bestimmten Verteilungen geringere erwartete Suchkosten als die Vertauschen-Regel erzeugt. Allerdings permutiert diese Regel auch Elemente, auf die nicht zugegriffen wurde. [3]*

Verschiedene Autoren haben die Menge der Regeln, die in Konkurrenz stehen, eingeschränkt. Die Klasse der **Vorwärts-Regeln** (*move-forward rules*) umfaßt alle Regeln, die das nachgefragte Element eine beliebige Schrittzahl in Richtung des Listenkopfs bewegt und die relative Ordnung aller anderen Elemente beibehält. Außerdem darf kein Element auf eine Position gesetzt werden, die hinter der Stelle liegt, an die ein ursprünglich weiter hinten liegendes Element gesetzt würde. Die oben beschriebenen Regeln *move-to-front, transposition, move-up(k), pos(k), switch(k)* gehören alle zu dieser Klasse.

Eine Vorwärts-Regel wird beschrieben durch n Zahlen $1 = m_1, m_2, m_3, \ldots m_n$ mit $m_j < j$ und $m_{j-1} \leq m_j$ für $j \geq 2$. Wenn auf das Element auf Position j zugegriffen wird, dann wird es auf Position m_j der Liste gesetzt.
Mit der oben erwähnten Darstellung durch Permutationen ist dann

$$\tau_j(k) = \left\{ \begin{array}{ccl} j & : & \text{falls } k = m_j \\ k-1 & : & \text{falls } m_j < k \leq j \\ k & : & \text{sonst} \end{array} \right\}$$

Auf diesen Regeln kann man eine partielle Ordnung einführen (Lam [25]):
Seien R und R' zwei Vorwärts-Regeln, definiert durch $1, m_2, \ldots, m_n$ bzw. $1, m'_2, \ldots, m'_n$. Dann definiert

$$R \leq R' \quad :\Longleftrightarrow \qquad m'_j \leq m_j \quad \forall j = 1, \ldots, n \tag{1.3}$$

eine partielle Ordnung auf den Vorwärts-Regeln.

Damit kann man folgendes zeigen (Kan/Ross [21], Lam [25], Phelps/Thomas [33], Tenenbaum/Nemes [36])

Satz 1.7:

> *Für die spezielle Klasse von Verteilungen*
>
> $$\begin{aligned} p_1 &= q \\ p_i &= (1-q)/(n-1) \text{ für } i = 2, \ldots, n \end{aligned}$$
>
> *gelten folgende Ergebnisse*
>
> 1. *die Vertauschen-Regel ist optimal innerhalb der Vorwärts-Regeln,*
> 2. $\mu(\text{MOVE-UP}(k)) \leq \mu(\text{MOVE-UP}(k+1))$
> 3. $\mu(\text{POS}(k+1)) \leq \mu(\text{POS}(k))$
> 4. $\mu(\text{SWITCH}(k)) \leq \mu(\text{SWITCH}(k+1))$
> 5. *die Nach-Vorne-Schieben-Regel besitzt die schlechteste erwartete Suchzeit unter allen Vorwärts-Regeln.*

Beweis:

Alle verglichenen Regeln erfüllen die oben definierte partielle Ordnung, z.B.

$$\text{MOVE-UP}(k) \leq \text{MOVE-UP}(k+1)$$

In [25] wird gezeigt, daß bei der angegebenen Verteilung für alle Vorwärts-Regeln R und R' gilt:

$$R \leq R' \implies \mu(R) \leq \mu(R')$$

■

1.3 Abhängige Zugriffe

Jetzt betrachten wir die Situation abhängiger Zugriffe. Von mehreren Autoren wurde angemerkt, daß in vielen Fällen die stochastische Modellierung der Nachfragen durch eine Markov-Kette sinnvoll ist. Der Zustandsraum der Kette entspricht der Menge S der gespeicherten Elemente. Die Anfragequelle besitzt ein Gedächtnis, dessen Inhalt durch den aktuellen Zustand $i \in S$ wiedergegeben wird. Er bestimmt die Wahrscheinlichkeitsverteilung für den nächsten Zugriff.

1.3.1 Optimale Strategien

Wenn man die Übergangsmatrix P der Markov-Kette kennt, dann kann man die Liste vor jedem Zugriff so umordnen, daß eine bestmögliche lineare Suche im aktuellen Zustand der Kette realisiert wird. Im Zustand i wird die Liste nach absteigenden Zugriffswahrscheinlichkeiten $p_{i\pi_i(1)} \geq p_{i\pi_i(2)} \geq \cdots \geq p_{i\pi_i(n)}$ geordnet, was durch die Permutation π_i der Elemente angegeben wird.
Wenn die Übergangsmatrix unbekannt ist, dann kann man n^2 viele Zähler benutzen und die Übergangswahrscheinlichkeiten p_{ij} durch relative Übergangshäufigkeiten annähern. Beide Verfahren sind asymptotisch optimal, wenn man die reinen Suchkosten betrachtet. Da aber vor jeder Suche die Liste entsprechend dem aktuellen Zustand umsortiert werden muß (was einer Suche nach allen Elementen gleichkommt), ist dieses Verfahren praktisch nicht durchführbar.
Von Konneker/Varol [24] wurde vorgeschlagen, für jeden Zustand einen Zeiger zu benutzen, der auf das Element mit der größten Übergangswahrscheinlichkeit zeigt.

Es scheint aussichtslos, ein Schema zu finden, das mit weniger als $\Omega(n^2)$ Platzbedarf auskommt und trotzdem für *alle* Markov-Ketten ein asymptotisch optimales Verhalten zeigt. Deshalb wurde vorgeschlagen, sich auf bestimmte Markov-Ketten zu konzentrieren.

1.3.2 Lokalität

Das Paging-Verhalten von Betriebssystemen zur Verwaltung von virtuellem Speicher läßt sich bis zu einem gewissen Grade durch Markov-Ketten erster Ordnung beschreiben. Hierbei tritt das Phänomen der *Lokalität* auf: wenige Speicherseiten werden längere Zeit ausschließlich genutzt, bevor auf eine andere kleine Teilmenge des gesamten Speichers zugegriffen wird und diese wieder für längere Zeit benutzt wird usw. (sogenannte *working sets*).

Ein anderes Beispiel für Lokalität entsteht bei der Benutzung eines mehrsprachigen Wörterbuchs (englisch-deutsch, französich-deutsch, spanisch-deutsch) in einem Band (Hotz [20]). Jede der Fremdsprachen bildet eine Teilmenge in der Gesamtmenge aller

fremdsprachigen Vokabeln. Beim Gebrauch des Wörterbuchs (z. B. zur Übersetzung von Texten ins Deutsche) wird man typischerweise über einen langen Zeitraum nur die Einträge einer Sprache benötigen, bevor man auf eine andere Sprache umschaltet.
Das Übergangsverhalten der Markov-Kette weist eine gewisse Trägheit auf.

Eine Anfragequelle mit Gedächtnis ist einer Umordnungs-Heuristik ohne Gedächtnis stets überlegen. Man kann zu jeder solchen Heuristik eine Anfragequelle konstruieren, deren erwartete Suchzeit beliebig nahe an der Suchzeit liegt, die man bei zufälliger Anordnung der Liste erzielt.
Wenn man aber nicht alle Anfragequellen zuläßt, sondern nur die, die das Phänomen der Lokalität aufweisen, dann können auch Umordnungs-Regeln ohne Gedächtnis zu guten Ergebnissen kommen. Dieser Gedanke ist grundlegend für die weiteren Analysen.

Mir sind nur zwei Artikel bekannt, die selbstorganisierende lineare Listen bei Zugriffsfolgen in Form einer Markov-Kette beschreiben. Beide Arbeiten widmen sich vor allem bzw. ausschließlich der Nach-Vorne-Schieben-Regel (Lam/Leung/Siu [26], Chassaing [9]).

1.3.3 Grundlagen der Analyse

Die Folge der Anfragen ist durch eine Markov-Kette P gegeben. Die verschiedenen Permutationen der linearen Liste unter der Nach-Vorne-Schieben-Regel bilden selbst eine Markov-Kette (siehe Seite 5). Wenn man das Gesamtsystem als Markov-Kette darstellen möchte, so erhält man Zustände

$$(\pi, i) \in \mathcal{S}_n \times \{1, \ldots, n\}.$$

Diese neugebildete, erweiterte Kette (*extended markov chain*) besitzt als Zustandsraum das Kreuzprodukt der Zustandsräume der beiden ursprünglichen Ketten (siehe auch Kapitel 4.2). Sie ist ergodisch, wenn P ergodisch ist, und die verwendete Heuristik den Übergang zwischen beliebigen Konfigurationen der Liste mit beschränkter Anzahl von Schritten erlaubt. Speziell gilt für die Vorwärts-Regeln (siehe [26]):

Satz 1.8:

> *Wenn P eine ergodische Markov-Kette ist, und eine Vorwärts-Regel verwendet wird, dann ist die erweiterte Markov-Kette auch ergodisch.*

Beweis:
Unter einer Vorwärts-Regel ist jede Konfiguration nach höchstens n^2 Schritten erreichbar, wenn eine beliebige Zugriffsfolge erlaubt ist. Es gibt ein n_0, so daß für alle $n \geq n_0$ alle Komponenten von P^n größer als Null sind. Nach den ersten n_0 Zugriffen ist also

jede Folge von Zugriffen möglich, und somit jede Konfiguration nach höchstens $n_0 + n^2$ Schritten erreichbar. Nach dieser Schrittzahl kann auch der nächste Zugriff auf jedes beliebige Element gerichtet sein. ■

Die Analyse der Nach-Vorne-Schieben-Regel gemäß Lam/Leung/Siu beruht auf zwei grundlegenden Begriffen aus der Theorie der Markov-Ketten:

Definition 1.9:

(a) *Rückwärtslaufende Ketten* (time-reversed chains)

Für ergodische Markov-Ketten kann man die Richtung der Zeit umkehren und die Kette „in die Vergangenheit“ laufen lassen. Denn die Wahrscheinlichkeit, von Zustand i einen Schritt rückwärts in den Zustand j zu machen, ist gleich der Wahrscheinlichkeit, sich in j zu befinden und von dort einen Schritt vorwärts nach i zu machen unter der Bedingung, sich jetzt in i zu befinden:

$$\hat{p_{ij}} = \frac{q_j p_{ji}}{q_i}.$$

Ab jetzt werden alle Größen, die sich auf rückwärtslaufende Markov-Ketten beziehen, mit einem Dach versehen. So ist $\hat{p_{ij}}$ die Übergangswahrscheinlichkeit, von i nach j rückwärts zu gehen, und $\hat{P}$ die Übergangsmatrix der rückwärtslaufenden Kette.

(b) *Tabu-Mengen von Zuständen* (taboo sets)

Eine Tabu-Menge $H \subseteq S$ ist eine Teilmenge der Zustandsmenge, die bei einer Folge von Übergängen nicht betreten werden darf.

Wie werden Übergangswahrscheinlichkeiten bei Anwesenheit von Tabu-Mengen berechnet?

Folgender Satz ist in (Kemperman [23]) angegeben.
Dabei bezeichnet m_{ij} die erwartete Erst-Eintritts-Zeit (*mean first passage time*) von i nach j. Das ist die erwartete Anzahl von Schritten, bis man zum ersten Mal j erreicht, wenn man in i startet.

Satz 1.10:

Sei P eine ergodische Markov-Kette mit Zustandsmenge S und Tabu-Menge $H \subseteq S$, und seien $i, j \in S$, $j \notin H$.
Die Wahrscheinlichkeit, von Zustand i nach Zustand j zu gelangen, ohne unterwegs einen Zustand aus H zu betreten, wird mit ${}_Hf^*_{ij}$ bezeichnet und es gilt

$$ {}_Hf^*_{ij} = \frac{\det((\delta_{ab}-1)m_{ab}+m_{ib}+m_{aj}-m_{ij})_{a,b\in H}}{\det((\delta_{ab}-1)m_{ab}+m_{jb}+m_{aj})_{a,b\in H}} $$

Der Satz gilt auch, wenn der Startzustand in der Tabu-Menge liegt: $i \in H$. Denn die Zustände aus H sind nur für die Zwischenschritte verboten. Für den Absprung aus i wird also die Verteilung $(p_{i\cdot})$ benutzt, danach wird i zum Tabu-Zustand.

Durch Einsetzen erhält man sofort folgendes Korollar.

Korollar 1.11:

Für einelementige Tabu-Mengen $H = \{i\}$ gilt

$$ {}_if^*_{kj} = \frac{m_{ki}+m_{ij}-m_{kj}}{m_{ij}+m_{ji}} $$

Man erhält Informationen über das aktuelle Aussehen der Datenstruktur, wenn man ihre Vergangenheit betrachtet. Mit dieser Überlegung kann man das Strukturlemma für die Nach-Vorne-Schieben-Regel umformulieren:

Lemma 1.12:

Wenn auf k zugegriffen wird, dann steht j vor i in der Liste genau dann, wenn wir bei k startend die Zugriffsfolge zurückverfolgen, und auf j treffen, bevor ein i aufgetreten ist.

Für die Analyse betrachten wir also eine rückwärtslaufende Kette mit dem Tabu-Zustand i. Gefragt ist nach der Wahrscheinlichkeit, in dieser Kette von k nach j zu gelangen.

In 1.1 haben wir für unabhängige Zugriffe definiert:

$b(j,i)$	:	asymptotische Wahrscheinlichkeit, daß j vor i steht
$\mu(i)$	:	erwartete Position von i
μ	:	erwartete Suchzeit bei der Nach-Vorne-Schieben-Regel.

Für abhängige Zugriffe benutzen wir

$b(j,i\|k)$	:	asymptotische Wahrscheinlichkeit, daß j vor i steht, wenn auf k zugegriffen wird
$\mu(i\|k)$	:	erwartete Position von i beim Zugriff auf k
μ	:	erwartete Suchzeit.

In Wirklichkeit benötigen wir nur die Größen $b(j,i|i)$ und $\mu(i|i)$, um die erwartete Suchzeit zu berechnen. Denn dafür brauchen wir die erwartete Position eines Elements nur dann, wenn auch darauf zugegriffen wird.

Lemma 1.13:

Es gilt

$$b(j,i|k) = \frac{\hat{m}_{ki} + \hat{m}_{ij} - \hat{m}_{kj}}{\hat{m}_{ij} + \hat{m}_{ji}} \tag{1.4}$$

und speziell

$$b(j,i|i) = \frac{m_{ii}}{m_{ij} + m_{ji}}. \tag{1.5}$$

Beweis:
Gemäß dem umgeformten Strukturlemma 1.12 suchen wir die Wahrscheinlichkeit, in der rückwärtslaufenden Kette von k nach j zu gelangen, ohne den Zustand i zwischendrin zu benutzen. Diese Wahrscheinlichkeit wird durch Korollar 1.11 gegeben, wenn man die Zeitumkehrung beachtet.

Im Spezialfall $i = k$ erhält man

$$b(j,i|i) = \frac{\hat{m}_{ii}}{\hat{m}_{ij} + \hat{m}_{ji}}$$

Es gilt aber (siehe Anhang A):

$$\begin{aligned} \hat{m}_{ii} &= m_{ii} \\ \hat{m}_{ij} + \hat{m}_{ji} &= m_{ij} + m_{ji} \end{aligned}$$

■

Obige Darstellung mit Hilfe von Satz 1.10 führt die Berechnung auf Werte aus der Theorie ergodischer Markov-Ketten zurück. Im nächsten Abschnitt beschreiben wir eine alternative Herleitung, die den für unsere Zwecke besser handhabbaren Ansatz absorbierender Markov-Ketten benutzt. Außerdem lassen sich die dabei auftretenden Werte leicht interpretieren.

Das Hauptergebnis bei Zugriffen in Form einer Markov-Kette ist folgender Satz.

Satz 1.14 (Lam/Leung/Siu 1984)
Die erwarteten Suchkosten unter der Nach-Vorne-Schieben-Regel sind

$$\mu = 1 + 2 \cdot \sum_{i<j} \frac{1}{m_{ij} + m_{ji}}. \tag{1.6}$$

Beweis:
Es gilt $m_{ii} = 1/q_i$: die erwartete Zeitdauer bis zur Rückkehr in den gleichen Zustand ist der Kehrwert der stationären Wahrscheinlichkeit dieses Zustands (siehe Anhang A). Die Berechnung verläuft analog zum Fall unabhängiger Zugriffe:

$$\begin{aligned}
\mu &= \sum_{i=1}^{n} q_i \cdot \mu(i|i) \\
&= \sum_{i=1}^{n} q_i \cdot (1 + \sum_{j\neq i} b(j,i|i)) \\
&= \sum_{i=1}^{n} q_i \cdot \left(1 + \sum_{j\neq i} \frac{1/q_i}{m_{ij} + m_{ji}}\right) \\
&= 1 + \sum_{i=1}^{n} \sum_{j\neq i} \frac{1}{m_{ij} + m_{ji}} \\
&= 1 + 2 \cdot \sum_{i<j} \frac{1}{m_{ij} + m_{ji}}
\end{aligned}$$

■

1.3.4 Beschreibung durch absorbierende Markov-Ketten

Die Berechnung von Übergangswahrscheinlichkeiten bei Anwesenheit von Tabu-Zuständen, die während der Übergänge nicht betreten werden dürfen, kann man auf absorbierende Markov-Ketten zurückführen:

Die Matrix P wird modifiziert, so daß alle Tabu-Zustände zu absorbierenden Zuständen werden. Ein Zustand i heißt **absorbierend**, wenn er nicht verlassen werden kann, d.h. $p_{ii} = 1$. Die i-te Zeile der Matrix ist dann die i-te Zeile der Einheitmatrix. Die so entstehende Markov-Kette ist nicht mehr ergodisch. Die Zustandsmenge zerfällt in zwei disjunkte Teilmengen:

- die *absorbierenden* Zustände H und

- die *transienten* Zustände T.

Die transienten Zustände werden irgendwann einmal verlassen und nicht wieder betreten. Die asymptotische Wahrscheinlichkeit, sich in einem transienten Zustand zu befinden, ist 0. Seien $r = |T|$ Zustände transient und $n-r = |H|$ Zustände absorbierend. Ohne Beschränkung der Allgemeinheit seien $H = \{1, \ldots, n-r\}$ und $T = \{n-r+1, \ldots, n\}$. Die ursprüngliche Übergangsmatrix habe folgende Gestalt

$$P = \left(\begin{array}{c|c} U & V \\ \hline R & Q \end{array} \right), \tag{1.7}$$

Die modifizierte Matrix $\tilde{P}$ hat dann folgende Gestalt:

$$\tilde{P} = \left(\begin{array}{c|c} I & O \\ \hline R & Q \end{array} \right), \tag{1.8}$$

wobei

I	:	Einheitsmatrix der Dimension $(n-r) \times (n-r)$
O	:	Null-Matrix der Dimension $(n-r) \times r$
$Q = P\vert_{T\times T}$	:	Übergangsmatrix der transienten Zustände.

Wie vorher kann man auch die Matrix Q iterieren. Für $m > 0$ ist $Q^m = (q_{ij}^{(m)})$ die Übergangsmatrix, in genau m Schritten von einem zu einem anderen transienten Zustand zu gelangen, ohne daß Absorption aufgetreten ist.
Für die Matrix Q gilt: $Q^m \to 0$ komponentenweise bei $m \to \infty$. (Die asymptotische Wahrscheinlichkeit, sich in einem transienten Zustand zu befinden, ist Null.)

Lemma 1.15 :

Die erwartete Anzahl von Besuchen des Zustands j bevor Absorption eintritt, wenn man in i startet ($i, j \in T$), ist gegeben durch die Einträge der Matrix

$$N = (n_{ij}) = I + Q + Q^2 + Q^3 + \cdots$$

Die Matrix N heißt Fundamentalmatrix der transienten Zustände, *und es gilt*

$$N = (I - Q)^{-1}.$$

Beweis:
Für $m > 0$ ist

$$I - Q^m = (I - Q) \cdot (I + Q + Q^2 + \cdots + Q^{m-1}).$$

Wegen $Q^m \to 0$ bei $m \to \infty$ hat die linke Seite für hinreichend großes m eine Determinante ungleich Null, und weil das Produkt von Determinanten gleich der Determinante des Produkts ist, auch die Matrizen auf der rechten Seite. Insbesondere existiert das Inverse von $I - Q$ und ist gleich N.

Die erwartete Anzahl von Besuchen in j bei Start in i erhält man durch Aufsummieren der Erwartungswerte in jedem Schritt. Die Startposition trägt 1 im Fall $i = j$ bei, 0 sonst, also δ_{ij}. Der nächste Schritt geht mit Wahrscheinlichkeit p_{ik} nach k. Er trägt nur etwas bei, wenn k transient ist, und zwar gerade n_{kj}. Zusammen:

$$\begin{aligned} n_{ij} &= \delta_{ij} + \sum_{k \in T} p_{ik} n_{kj} \\ N &= I + Q \cdot N, \text{ also} \\ N &= (I - Q)^{-1} \end{aligned}$$

■

Satz 1.16:

Sei $H \subseteq S$ eine Tabu-Menge und $i, j \in H$. Die Wahrscheinlichkeit, von i nach j zu gelangen, ohne zwischendrin *einen Zustand aus H zu benutzen, ist gegeben durch den Eintrag c_{ij} der Matrix*

$$C = (c_{ij}) = U + VNR. \tag{1.9}$$

Beweis:

Die Zustände aus H sollen nur als Zwischenzustände verboten sein. Zum Start von i aus betrachten wir die alte Situation der Zeile i in der Matrix P. Sobald i dann verlassen ist, wird er absorbierend. Für die nun folgenden Übergänge in den transienten Zuständen wird die Matrix Q benutzt. Am Ende tritt Absorption in den speziellen Zustand j ein.

Entweder wir springen direkt von i nach j, oder wir springen zu einem transienten Zustand und verbleiben einige Zeit in den transienten Zuständen und werden schließlich in j absorbiert. Der Einsprung in den transienten Zustand k geschieht mit Wahrscheinlichkeit p_{ik}. Immer wenn man einen transienten Zustand k' erreicht hat, kann man von dort mit Wahrscheinlichkeit $p_{k'j}$ abspringen. Gemäß Lemma 1.15 gibt es im Mittel $n_{kk'}$ Zeitpunkte für den Absprung von k'. Da für verschiedene Paare von Ein- und Absprungzuständen (k, k') disjunkte Ereignisse vorliegen, kann man die entsprechenden Wahrscheinlichkeiten aufaddieren:

$$c_{ij} = p_{ij} + \sum_{k \in T} \sum_{k' \in T} p_{ik} n_{kk'} p_{k'j} \tag{1.10}$$

Wenn man $i, j \in H$ und $k, k' \in T$ berücksichtigt, kann man dies mit Hilfe der entsprechenden Teilmatrizen aus (1.7) auch so schreiben:

$$C = U + VNR.$$

■

Für die Analyse der Nach-Vorne-Schieben-Regel muß nur noch die Tabu-Menge richtig interpretiert werden. Nach dem Strukturlemma 1.12 gibt es zwei Tabuzustände:

$$H = \{i, j\} \tag{1.11}$$

Korollar 1.17 :

Für die asymptotische Wahrscheinlichkeit $b(j, i|i)$, daß das Element j vor i steht, gilt mit Satz 1.16 und $H = \{i, j\}$

$$b(j, i|i) \quad = \quad \hat{c_{ij}} \tag{1.12}$$

1.3.5 Beispiele

Im folgenden werden einige spezielle Markov-Ketten als Modellierung der Anfrage-Quelle betrachtet und die erwartete Suchzeit unter der Nach-Vorne-Schieben-Regel berechnet.

1.3.5.1 Beispiel 1

Ein sehr einfaches Beispiel für abhängige Zugriffe mit Lokalität wird durch die Übergangsmatrix $P = (p_{ij})$ der Dimension $n = \dim(P)$ gegeben:

$$p_{ij} = \left\{ \begin{array}{ccc} \alpha & : & i = j \\ \beta & : & i \neq j \end{array} \right\} \tag{1.13}$$

mit $\beta = (1 - \alpha)/(n - 1)$, und $0 \leq \alpha < 1$.
Hier ist also die Wahrscheinlichkeit, auf das zuletzt nachgefragte Element nochmals zuzugreifen, gleich α, und die Wahrscheinlichkeit, ein bestimmtes anderes Element nachzufragen, gleich β. Die Bedingung $\alpha < 1$ stellt sicher, daß P die Übergangsmatrix einer ergodischen Markov-Kette ist.

Die Matrix P ist symmetrisch und damit eine *doppelt stochastische Matrix*: auch die Spaltensummen sind 1. Zusammen ergibt sich, daß die Markov-Kette zeit-umkehrbar (*time-reversible*) ist, d.h. die Übergangsmatrix $\hat{P}$ für die rückwärtslaufende Kette ist gleich P. Das liegt daran, daß bei einer doppelt stochastischen Übergangsmatrix die stationäre Verteilung gerade die Gleichverteilung ist. In diesem Fall gilt für die Übergänge $\hat{p}_{ij}$ der rückwärtslaufenden Kette

$$\hat{p}_{ij} = \frac{1/n \cdot p_{ji}}{1/n} = p_{ji},$$

also $\hat{P} = P^\top$, und da die Matrix symmetrisch ist, auch $\hat{P} = P$.

Wir wollen die erwartete Suchzeit unter der Nach-Vorne-Schieben-Regel mit Hilfe von Korollar 1.17 ausrechnen.
Für jede Wahl eines Paares $i \neq j$ von Zuständen betrachten wir die Tabu-Menge $H = \{i, j\}$ und die Menge der transienten Zustände $T = \{1, \ldots, n\} - H$. Die Übergangsmatrix Q der transienten Zustände entsteht aus P, indem die Zeilen und Spalten i und j gestrichen werden. Aufgrund der einfachen Struktur von P ist Q für alle Wahlen von i, j gleich.
Sei $r = \dim(Q) = n-2$, und sei $N = (I-Q)^{-1}$ die Fundamentalmatrix der transienten Zustände.
Man berechnet N, indem man den Ansatz $n_{ii} = a$ und $n_{ij} = b$ für alle $i \neq j$ macht, und dann das folgende 2×2-System für a und b löst:

$$\begin{aligned} (1-\alpha)a - (r-1)\beta b &= 1 \\ -\beta a + ((1-\alpha) - (r-2)\beta)b &= 0 \end{aligned}$$

Für die Spaltensummen von N gilt für jede Spalte k':

$$\begin{aligned} \sum_{k \in T} n_{kk'} &= \frac{1}{1-\alpha+\beta(1-r)} \\ &= \frac{1}{2\beta} \end{aligned}$$

Damit erhalten wir für die asymptotische Wahrscheinlichkeit, daß j vor i steht beim Zugriff auf i:

$$\begin{aligned} b(j, i|i) &= p_{ij} + \sum_{k \in T} \sum_{k' \in T} p_{ik} n_{kk'} p_{k'j} \\ &= \beta + \beta \cdot \frac{r}{2\beta} \cdot \beta \\ &= \beta \cdot (1 + r/2) \\ &= \beta n/2 \end{aligned}$$

und für die erwartete Suchzeit

$$\mu = \sum_i \mu(i) \cdot 1/n$$

$$
\begin{aligned}
&= \sum_i \Big(1 + \sum_{j \neq i} b(j,i|i)\Big) \cdot 1/n \\
&= 1 + \sum_i (n-1) \cdot (\beta n/2) \cdot 1/n \\
&= 1 + (1-\alpha) \cdot n/2
\end{aligned}
$$

Satz 1.18:

> *Wenn die Zugriffsfolge durch die Markov-Kette (1.13) gegeben ist, dann gilt für die erwartete Suchzeit unter der Nach-Vorne-Schieben-Regel*
>
> $$\mu = 1 + (1-\alpha) \cdot n/2 \tag{1.14}$$

Interpretation:
Im Grenzfall $\alpha \to 1$ erhält man $\mu = 1$. Nach höchstens einem Schritt befindet sich das nachgefragte Element an erster Position. Dort wird es immer verbleiben, die Suche ist also stets nach einem Vergleich beendet.
Bei $\alpha = 1/n$ ergibt sich $\mu = (n+1)/2$, die erwartete Suchzeit bei zufälliger Ordnung der Liste.
Bei $\alpha = 0$ wird das gleiche Element niemals zweimal hintereinander nachgefragt. Es gilt: $\mu = 1 + n/2 = (n+1)/2 + 1/2$. Verglichen mit zufälliger statischer Anordnung der Liste sieht man, daß man sich für die nächste Suche ein Element in den Weg legt, was bei der Hälfte der Elemente (erste Hälfte der Liste) nicht dagewesen wäre. Deshalb ist die erwartete Suchzeit um 1/2 größer als bei zufälliger Anordnung.

Für $\alpha \geq \beta$, hier also $\alpha \geq 1/n$, liegt das Phänomen der Lokalität vor. Die stationäre Verteilung ist die Gleichverteilung, aber durch die Trägheit der Übergänge wird die erwartete Suchzeit verbessert, und zwar in linearer Abhängigkeit vom Parameter α.

1.3.5.2 Beispiel 2

Wir wollen das erste Beispiel verallgemeinern, so daß als stationäre Verteilung nicht nur die Gleichverteilung, sondern jede beliebige vorgegebene Verteilung angenommen wird. Auch in diesem Fall kann die erwartete Suchzeit beliebig nah an 1 kommen, wenn die Trägheit der Übergänge groß genug ist.
Dieses Beispiel wurde von Lam/Leung/Siu [26] betrachtet.

Sei $\vec{p} = (p_1, \ldots, p_n)$ eine Wahrscheinlichkeitsverteilung, $p_i > 0$, und A die Übergangsmatrix mit den Zeilen $\vec{p}$, bei der also die Übergangswahrscheinlichkeiten unabhängig vom Zustand sind.
Sei weiter λ ein Parameter, $0 < \lambda \leq 1$. Wir betrachten die Matrix

$$P = \lambda A + (1-\lambda)I. \tag{1.15}$$

Dann ist P die Übergangsmatrix einer ergodischen Markov-Kette. Die stationäre Verteilung $\vec{q}$ ist gleich der Verteilung $\vec{p}$. Denn für beliebiges q_j gilt:

$$\begin{aligned} q_j &= \sum_i q_i p_{ij} \\ &= \sum_i q_i(\lambda p_j + (1-\lambda)\delta_{ij}) \\ &= \lambda p_j \sum_i q_i + q_j(1-\lambda) \\ &= \lambda p_j + q_j(1-\lambda) \end{aligned}$$

Durch Umstellen und Division durch $\lambda > 0$ folgt: $q_j = p_j$.

Wir berechnen die Matrix M der Erst-Eintritts-Zeiten, um Satz 1.14 anwenden zu können. Sie ergibt sich zu

$$m_{ij} = (\delta_{ij} - z_{ij} + z_{jj})/q_j,$$

wobei $Z = (I-P+Q)^{-1} = I + \sum_{m=1}^{\infty}(P^m - Q)$ die Fundamentalmatrix der ergodischen Kette P ist (siehe Anhang A). Hierbei ist Q die Grenzmatrix zu P^m, ihre Zeilen sind die Verteilungen $\vec{q}$. In obigem Beispiel ist also $Q = A$. Die Matrix Q ist idempotent: $Q^2 = Q$.

Für P^m gilt folgende Rekursionsgleichung:

$$P^m = a^{(m)} \cdot Q + b^{(m)} \cdot I$$

mit den Koeffizienten

$$a^{(1)} = \lambda, \qquad b^{(1)} = 1 - \lambda$$

sowie (beachte $Q^2 = Q$):

$$\begin{aligned} P^{m+1} &= P \cdot P^m \\ &= (\lambda Q + (1-\lambda)I) \cdot \left(a^{(m)}Q + b^{(m)}I\right) \\ &= \underbrace{(\lambda a^{(m)} + (1-\lambda)a^{(m)} + \lambda b^{(m)})}_{a^{(m+1)}} \cdot Q + \underbrace{(1-\lambda)b^{(m)}}_{b^{(m+1)}} \cdot I \end{aligned}$$

Die Koeffizienten lassen sich geschlossen darstellen:

$$\begin{aligned} b^{(m+1)} &= (1-\lambda)^{m+1} \\ a^{(m+1)} &= a^{(m)} + \lambda(1-\lambda)^m \\ &= \lambda \sum_{j=0}^{m} (1-\lambda)^j \\ &= \lambda \frac{1-(1-\lambda)^{m+1}}{1-(1-\lambda)} \\ &= 1 - (1-\lambda)^{m+1}, \end{aligned}$$

und man erhält
Lemma 1.19:

Für die Potenzen der Matrix $P = \lambda Q + (1-\lambda)I$, $0 < \lambda \le 1$, *gilt*

$$\begin{aligned} P^m &= (1-(1-\lambda)^m)\cdot Q + (1-\lambda)^m \cdot I \\ &= Q + (1-\lambda)^m (I - Q) \end{aligned} \tag{1.16}$$

Damit kann man die Fundamentalmatrix Z berechnen:

$$\begin{aligned} Z &= I + \sum_{m=1}^{\infty} (P^m - Q) \\ &= I + (I-Q)\sum_{m=1}^{\infty}(1-\lambda)^m \\ &= I + (I-Q)(1/\lambda - 1) \\ &= (I - (1-\lambda)Q)/\lambda \end{aligned}$$

sowie die Matrix der Erst-Eintritts-Zeiten M:

$$\begin{aligned} m_{ij} &= (\delta_{ij} - z_{ij} + z_{jj})/q_j \\ &= (\delta_{ij} - (\delta_{ij} - (1-\lambda)q_j)/\lambda + (1-(1-\lambda)q_j)/\lambda)/q_j \\ &= \frac{1 + \delta_{ij}(\lambda - 1)}{\lambda q_j} \end{aligned}$$

Gemäß Satz 1.14 ist dann

$$\begin{aligned} \mu(P) &= 1 + 2\sum_{i<j}\frac{1}{m_{ij} + m_{ji}} \\ &= 1 + 2\lambda\sum_{i<j}\frac{q_i q_j}{q_i + q_j} \\ &= \lambda \cdot \mu(A) + (1-\lambda) \end{aligned} \tag{1.17}$$

wenn man mit $\mu(A)$ die erwartete Suchzeit bezeichnet, die die Nach-Vorne-Schieben-Regel bei unabhängigen Zugriffen mit der Verteilung $\vec{q}$ bezeichnet (vgl. Formel (1.1)).

Satz 1.20:

Für die Zugriffsfolge (1.15) ist die erwartete Suchzeit $\mu(P)$ *unter der Nach-Vorne-Schieben-Regel*

$$\mu(P) = \lambda \cdot \mu(A) + (1-\lambda) \tag{1.18}$$

Das Beispiel 1 ergibt sich als Spezialfall, wenn man $\vec{p} = (1/n, \ldots, 1/n)$ und

$\lambda = (1-\alpha)n/(n-1)$ setzt. Dann ist $p_i p_j/(p_i + p_j) = 1/2n$ und

$$\begin{aligned} \mu &= 1 + 2\lambda \cdot n(n-1)/2 \cdot 1/2n \\ &= 1 + (1-\alpha)n/(n-1) \cdot (n-1)/2 \\ &= 1 + (1-\alpha)n/2 \end{aligned}$$

Satz 1.21 (Lam/Leung/Siu 1984)
Für eine beliebige Wahrscheinlichkeitsverteilung $\vec{p} = (p_1, \ldots, p_n)$, $p_i > 0$ und jedes $\varepsilon > 0$ existiert eine Markov-Kette P, so daß $\vec{p}$ die stationäre Verteilung der Kette ist und für die erwartete Suchtzeit μ unter der Nach-Vorne-Schieben-Regel gilt

$$\mu < 1 + \varepsilon \tag{1.19}$$

Beweis:
Sei A die Übergangsmatrix, deren Zeilen $\vec{p}$ sind.
Wähle λ so, daß $0 < \lambda < \min(1, \varepsilon/(\mu(A) - 1))$, und setze $P = \lambda A + (1-\lambda)I$ gemäß Formel (1.15). Nach Formel (1.17) und der Wahl von λ folgt:

$$\begin{aligned} \mu &= \lambda\mu(A) + (1-\lambda) \\ &= 1 + \lambda(\mu(A) - 1) \\ &< 1 + \varepsilon \end{aligned}$$

■

Bei diesem Beispiel kommt die Trägheit der Übergänge dadurch zustande, daß die Wahrscheinlichkeit, beim gleichen Element zu bleiben, erhöht wird, während die Wahrscheinlichkeit eines Wechsels verkleinert wird. Die Hauptdiagonalelemente der Übergangsmatrix werden angehoben, und die anderen Elemente werden gesenkt. Wir wollen dies als *elementweise Lokalität* bezeichnen.

1.3.5.3 Beispiel 3

Das vorhergehende Beispiel soll weiter verallgemeinert werden, und zwar so, daß die Lokalität der Zugriffe nicht elementweise auftritt, sondern sich über Teilmengen erstreckt (*locality by bunches*). Als stationäre Verteilung soll wieder jede vorgegebene Verteilung angenommen werden können.

Eine Variante dieses Beispiels wurde von Bellow [6] untersucht. Dort sind alle Übergänge zwischen Teilmengen gleich wahrscheinlich, und die Aufenthaltsdauer in einer Teilmenge ist exponentialverteilt. Die Betrachtung ist von dem LRU-Verfahren (*least recently used*) zur Seitenspeicherverwaltung und dem Working Set Model inspiriert.

Wir benutzen einen anderen Ansatz, der die Analogie zum vorherigen Beispiel deutlich werden läßt.

Da bei sequentieller Suche keine Ordnung auf den Elementen benutzt wird, kann man ohne Beschränkung der Allgemeinheit eine Partitionierung

$$S = \{1, \ldots, k_1\} \cup \{k_1 + 1, \ldots, k_2\} \cup \ldots \cup \{k_{w-1} + 1, \ldots, n\}$$

mit $0 = k_0 < k_1 < k_2 < \cdots < k_{w-1} < k_w = n$ annehmen. Wir bezeichnen die Teilmengen mit $X_i, i = 1, \ldots, w$:

$$X_i = \{k_{i-1} + 1, \ldots, k_i\}.$$

Sei eine Wahrscheinlichkeitsverteilung $\vec{p} = (p_1, \ldots, p_n)$ gegeben, und sei A die Matrix der unabhängigen Übergänge, bei der also alle Zeilen aus der Verteilung $\vec{p}$ bestehen.

Wir definieren für $i = 1, \ldots, w$ die Summe der Wahrscheinlichkeiten jeder Teilmenge

$$s_i = \sum_{j \in X_i} p_j.$$

Die Matrix J sei

$$J = \left(\begin{array}{c|c|c|c} \frac{1}{s_1} A|_{X_1 \times X_1} & & & \\ \hline & \frac{1}{s_2} A|_{X_2 \times X_2} & & \\ \hline & & \ddots & \\ \hline & & & \frac{1}{s_w} A|_{X_w \times X_w} \end{array} \right)$$

wobei $A|_{X_i \times X_i}$ die Einschränkung von A auf die Zeilen und Spalten aus X_i bedeutet. Durch die Gewichtung mit dem Faktor $1/s_i$ wird erreicht, daß jede Zeilensumme von J gerade 1 ist.

Für $0 < \lambda \leq 1$ definieren wir

$$P = \lambda A + (1 - \lambda) J \tag{1.20}$$

Dann ist P die Übergangsmatrix einer ergodischen Markov-Kette.
Die stationäre Verteilung $\vec{q}$ ist gleich der Verteilung $\vec{p}$. Denn für beliebiges q_j mit $j \in X_l$ gilt:

$$\begin{aligned} q_j &= \sum_i q_i p_{ij} \\ &= \lambda p_j \cdot \sum_i q_i \; + \; (1-\lambda)(p_j/s_l) \cdot \sum_{i \in X_l} q_i \\ &= p_j \Big(\lambda + ((1-\lambda)/s_l) \sum_{i \in X_l} q_i\Big) \end{aligned} \tag{1.21}$$

Durch Aufsummieren über alle $j \in X_l$ erhält man

$$\begin{aligned} \sum_{j \in X_l} q_j &= s_l \Big(\lambda + (1-\lambda)/s_l \sum_{i \in X_l} q_i\Big) \\ &= \lambda s_l + (1-\lambda) \sum_{i \in X_l} q_i \end{aligned}$$

und daraus wegen $\lambda \neq 0$

$$\sum_{j \in X_l} q_j = s_l$$

Durch Einsetzen in (1.21) folgt: $q_j = p_j$. Also ist auch die Grenzmatrix Q gleich A.

Wir berechnen die Fundamentalmatrix Z und die Matrix M der Erst-Eintritts-Zeiten.

Man sieht, daß $Q^2 = Q$ und $J^2 = J$ idempotent sind, und daß sich J zu Q wie ein neutrales Element verhält: $QJ = JQ = Q$. Deshalb kann der Beweis von Lemma 1.19 unverändert übernommen werden, und man erhält

$$P^m = Q + (1-\lambda)^m (J - Q). \tag{1.22}$$

Ebenso gilt für die Fundamentalmatrix

$$Z = I + (J - Q)(1/\lambda - 1).$$

Für die mittleren Erst-Eintritts-Zeiten machen wir eine Fallunterscheidung.
FALL 1: beide Indizes liegen in einer Teilmenge: $i, j \in X_l$

$$\begin{aligned} m_{ij} &= (\delta_{ij} - z_{ij} + z_{jj})/q_j \\ &= (\delta_{ij} - (\delta_{ij} + (q_j/s_l - q_j)(1/\lambda - 1)) + (1 + (q_j/s_l - q_j)(1/\lambda - 1)))/q_j \\ &= 1/q_j \end{aligned}$$

FALL 2: die Indizes liegen in verschiedenen Teilmengen: $j \in X_l, i \notin X_l$

$$\begin{aligned} m_{ij} &= (\delta_{ij} - z_{ij} + z_{jj})/q_j \\ &= (\delta_{ij} - (\delta_{ij} + (-q_j)(1/\lambda - 1)) + (1 + (q_j/s_l - q_j)(1/\lambda - 1)))/q_j \\ &= (1 + q_j/s_l(1/\lambda - 1))/q_j \\ &= \frac{\lambda(s_l - q_j) + q_j}{\lambda s_l q_j} \end{aligned}$$

Satz 1.22:

Wenn mit $\mu(Q)$ die erwartete Suchzeit bei unabhängigen Zugriffen mit der stationären Verteilung $\vec{q}$ bezeichnet wird, und $\sum^*$ die Summation über alle Paare von Elementen in verschiedenen Teilmengen X_k und X_l, $k \neq l$ ist, dann gilt für die erwartete Suchzeit

$$\mu(P) = \mu(Q) - {\sum}^*_{\substack{i \in X_k \\ j \in X_l}} \frac{(1-\lambda)(s_k + s_l) q_i q_j}{\lambda s_k s_l (q_i + q_j) + (1-\lambda)(s_k + s_l) q_i q_j} \cdot \frac{q_i q_j}{q_i + q_j} \quad (1.23)$$

Beweis:
Für $i, j \in X_l$ (Fall 1) ist

$$\frac{1}{m_{ij} + m_{ji}} = \frac{q_i q_j}{q_i + q_j}$$

Für $i \in X_k$, $j \in X_l$ mit $k \neq l$ (Fall 2) ist

$$\begin{aligned} \frac{1}{m_{ij} + m_{ji}} &= \frac{\lambda s_k s_l q_i q_j}{\lambda s_k s_l (q_i + q_j) + (1-\lambda) q_i q_j (s_k + s_l)} \\ &= \frac{q_i q_j}{q_i + q_j} \cdot \left(1 - \frac{(1-\lambda)(s_k + s_l) q_i q_j}{\lambda s_k s_l (q_i + q_j) + (1-\lambda)(s_k + s_l) q_i q_j}\right) \end{aligned}$$

Wenn mit $\mu(Q)$ die erwartete Suchzeit bei unabhängigen Zugriffen mit der Verteilung $\vec{q}$ bezeichnet wird,

$$\mu(Q) = 1 + \sum_i \sum_{j \neq i} \frac{q_i q_j}{q_i + q_j},$$

dann ist also

$$\mu(P) = \mu(Q) - \sum_{k=1}^{w} \sum_{\substack{l=1 \\ l \neq k}}^{w} \sum_{i \in X_k} \sum_{j \in X_l} \frac{(1-\lambda)(s_k + s_l) q_i q_j}{\lambda s_k s_l (q_i + q_j) + (1-\lambda)(s_k + s_l) q_i q_j} \cdot \frac{q_i q_j}{q_i + q_j}$$

■

Im Spezialfall $w = n$ (d.h. nur einelementige Teilmengen) erhält man wieder die Ergebnisse von Beispiel 2. Für $w < n$ gilt jedoch nicht mehr das Resultat von Satz 1.21, daß durch Verstärkung der Lokalität die erwartete Suchzeit beliebig nahe an 1 liegen kann. Statt dessen gilt

Satz 1.23:

Für alle Wahlen von λ, $0 < \lambda \leq 1$ gilt

$$\mu > 1 + \sum_{l=1}^{w} \sum_{i \in X_l} \sum_{\substack{j \in X_l \\ j \neq i}} \frac{q_i q_j}{q_i + q_j}$$

Beweis:
Wir betrachten den Quotient $1/(m_{ij} + m_{ji})$ im Fall 2 als Funktion von λ und berechnen die Ableitung.

$$\begin{aligned}
f(\lambda) &= \frac{\lambda s_k s_l q_i q_j}{\lambda s_k s_l (q_i + q_j) + (1-\lambda) q_i q_j (s_k + s_l)} \\
f'(\lambda) &= \frac{q_i^2 q_j^2 s_k s_l (s_k + s_l)}{(\lambda s_k s_l (q_i + q_j) + (1-\lambda) q_i q_j (s_k + s_l))^2}
\end{aligned}$$

Die Ableitung ist für alle λ positiv. Im Intervall $[0, 1]$ nimmt die Funktion an der Stelle $\lambda = 0$ den kleinsten Wert $f(0) = 0$ an. Also ist 0 eine untere Schranke für den Quotient im Fall 2. Für $\lambda > 0$ erhält man für die erwartete Suchzeit

$$\begin{aligned}
\mu(P) &= 1 + \sum_{\substack{i \neq j \\ \text{Fall 1}}} \frac{1}{m_{ij} + m_{ji}} + \sum_{\substack{i \neq j \\ \text{Fall 2}}} \frac{1}{m_{ij} + m_{ji}} \\
&> 1 + \sum_{\substack{i \neq j \\ \text{Fall 1}}} \frac{1}{m_{ij} + m_{ji}}
\end{aligned}$$

■

Spezialfall:
Wir illustrieren dieses Beispiel am Fall $\vec{q} = (1/n, \ldots, 1/n)$ und $|X_l| = k$ für alle l. Wenn die stationäre Verteilung die Gleichverteilung ist, und alle Teilmengen X_l die feste Größe k haben, gilt für die Wahrscheinlichkeitssummen $s_l = k/n$. Wir setzen

$$p_{ij} = \left\{ \begin{array}{lll} \alpha & : & i \text{ und } j \text{ liegen in gleicher Teilmenge} \\ \beta & : & \text{sonst} \end{array} \right\},$$

so daß gilt

$$\lambda = n(1 - \alpha k)/(n - k).$$

Wenn i und j in gleicher Teilmenge liegen, dann ist

$$\frac{1}{m_{ij} + m_{ji}} = \frac{1}{2n},$$

was $n(k-1)$ mal in der Summe für μ auftaucht. Im zweiten Fall ist

$$\frac{1}{m_{ij} + m_{ji}} = \frac{1 - \alpha k}{2(n-1) - 2\alpha n(k-1)},$$

was $n(n-k)$ mal aufsummiert wird. Zusammen ergibt sich

$$\begin{aligned} \mu &= 1 + \frac{n(k-1)}{2n} + \frac{n(n-k)(1-\alpha k)}{2(n-1) - 2\alpha n(k-1)} \\ &= 1 + \frac{k-1}{2} + \frac{n}{2} \cdot \frac{(n-k) - \alpha k(n-k)}{(n-1) - \alpha n(k-1)} \end{aligned} \tag{1.24}$$

Für $k = 1$ erhält man das Ergebnis von Beispiel 1: $\mu = 1 + (1-\alpha)n/2$.
Für beliebige k kann folgende Interpretation gemacht werden:
Im Grenzfall $\alpha \to 1/k$ erhält man $\mu = 1 + (k-1)/2$, also die erwartete Suchzeit unter k Elementen bei zufälliger Anordnung. Die Anfragen sind auf eine feste Teilmenge beschränkt, innerhalb der Gleichverteilung herrscht. Die Aussage von Satz 1.23 ergibt hier:
Für alle $0 \le \alpha < 1/k$ gilt:

$$\mu > 1 + \frac{k-1}{2}$$

Bei $\alpha = 1/n$ ergibt sich wieder $\mu = (n+1)/2$, die erwartete Suchzeit bei zufälliger Anordnung der Liste.
Bei diesem Beispiel gilt, was man erwartet: im Gegensatz zur elementweisen Lokalität von Beispiel 2 ist die Trägheit zwischen Teilmengen nicht groß genug, um beliebig kleine Suchzeiten zu erzielen. Trotzdem kann in diesem Fall die mittlere Suchzeit bei der Nach-Vorne-Schieben-Regel optimal sein (siehe Abschnitt 1.3.6).

1.3.5.4 Beispiel 4

Die Anfragequelle kann Zugriffsfolgen produzieren, die zu sehr schlechter erwarteter Suchzeit führen. Denn da die Heuristik kein Gedächtnis besitzt, ist eine Anfragequelle mit internem Zustand ihr stets überlegen.

Wenn die Zugriffsfolge zyklisch ist, entsteht bei der Nach-Vorne-Schieben-Regel nach spätestens einem Zyklus die Situation, daß immer auf das letzte Element der Liste

zugegriffen wird. Die asymptotische Suchzeit ist also n, die Länge der Liste.
Eine zyklische Zugriffsfolge ist jedoch nicht mehr ergodisch, also würde obige Analyse zusammenbrechen. Wenn man aber Folgen betrachtet, die „fast immer“ zyklisch sind, so kann die erwartete Suchzeit beliebig nahe an n liegen. Folgendes Beispiel wurde von Lam/Leung/Siu [26] angegeben.
Sei R die folgende Übergangsmatrix einer zyklischen Markov-Kette.

$$R = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & & \ddots & \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

Definiere für $0 < \alpha < 1$ die Matrix

$$P = \alpha R + (1 - \alpha) I. \tag{1.25}$$

Dann ist P die Übergangsmatrix einer ergodischen Markov-Kette mit stationärer Verteilung $\vec{q} = (1/n, \ldots, 1/n)$.

Satz 1.24 (Lam/Leung/Siu 1984)
Bei der durch P gemäß Formel (1.25) definierten Zugriffsfolge erzeugt die Nach-Vorne-Schieben-Regel die erwartete Suchzeit

$$\mu = 1 + \alpha(n - 1).$$

Mit $\alpha \to 1$ ist $\mu \to n$.

Beweis:
Für die Matrix M der mittleren Erst-Eintritts-Zeiten gilt

$$M = \frac{1}{\alpha} \cdot \begin{pmatrix} n & 1 & 2 & \cdots & n-1 \\ n-1 & n & 1 & \cdots & n-2 \\ n-2 & n-1 & n & \cdots & n-3 \\ & & & \ddots & \\ 1 & 2 & 3 & \cdots & n \end{pmatrix} - \frac{1-\alpha}{\alpha} \cdot n \cdot I.$$

Dies überprüft man, indem man nachrechnet, daß M die Beziehung

$$m_{ij} = 1 - p_{ij} m_{jj} + \sum_k p_{ik} m_{kj}$$

erfüllt. Die Matrix der mittleren Erst-Eintritts-Zeiten ist die eindeutige Matrix M, für die dies gilt (siehe Anhang A).
Wegen $m_{ij} + m_{ji} = n/\alpha$ für $i \neq j$ folgt

$$\begin{aligned} \mu &= 1 + \sum_i \sum_{j \neq i} \frac{\alpha}{n} \\ &= 1 + \alpha(n-1) \end{aligned}$$

■

Wie von Lam/Leung/Siu weiter bemerkt wurde, ist bei diesem schlechten Abschneiden der Nach-Vorne-Schieben-Regel wesentlich, daß die stationäre Verteilung die Gleichverteilung ist. Sie haben gezeigt, daß bei einer stationären Verteilung nach Zipf die erwartete Suchzeit mit der Nach-Vorne-Schieben-Regel stets besser ist als eine zufällige statische Anordnung der Elemente.

Satz 1.25 (Lam/Leung/Siu 1984)
Sei P eine beliebige ergodische Markov-Kette mit stationärer Verteilung $q_i = 1/(iH_n)$ (Verteilung nach Zipf). Dabei ist $H_n = 1 + 1/2 + \cdots + 1/n$ die n-te harmonische Zahl. Dann gilt für $n \geq 16$:

$$\mu < (n+1)/2.$$

Beweis:
Wegen $b(j,i|i) = 1/(q_i(m_{ij} + m_{ji}))$ und $b(j,i|i) \leq 1$ ist

$$\frac{1}{m_{ij} + m_{ji}} \leq q_i$$

Analog gilt $1/(m_{ji} + m_{ij}) \leq q_j$, zusammen also

$$\frac{1}{m_{ij} + m_{ji}} \leq \min(q_i, q_j)$$

Unter der Annahme $q_1 \geq q_2 \geq \cdots \geq q_n$ ergibt Einsetzen in Formel 1.6

$$\begin{aligned} \mu &\leq 1 + 2 \sum_{i<j} \min(q_i, q_j) \\ &= 1 + 2\Big(q_2 + 2q_3 + \cdots + (n-1)q_n\Big) \\ &= 1 + 2\Big(q_1 + 2q_2 + \cdots + nq_n \; - \; (q_1 + q_2 + \cdots + q_n)\Big) \\ &= 1 + 2(M-1) \\ &= 2 \cdot M - 1 \end{aligned}$$

wenn man mit M die optimale Suchzeit bei statischer Listenordnung bezeichnet. Für die stationäre Verteilung nach Zipf, $q_i = 1/(iH_n)$, gilt $M = n/H_n$ und damit

$$\mu \leq 2n/H_n - 1$$

Man sieht, daß für $n \geq 16$ die Behauptung folgt. ∎

1.3.6 Vergleich von Regeln

In diesem Abschnitt wird auf die Frage eingegangen, wie sich verschiedene Regeln hinsichtlich der asymptotischen Suchzeit unter abhängigen Zugriffen unterscheiden.

Zu dieser Fragestellung ist mir nur die Arbeit von Chassaing [9] bekannt. Er konnte zeigen, daß für gewisse Klassen von Markov-Ketten mit Lokalität die erwartete Suchzeit bei der Nach-Vorne-Schieben-Regel optimal ist. Optimal bezieht sich hier auf alle bisher untersuchten Verfahren zur Organisation von Listen.
Zur Konkurrenz zugelassen sind nicht nur die Regeln ohne Gedächtnis, sondern alle Regeln, die die gesamte bisherige Zugriffsfolge für ihre Entscheidung benutzen dürfen. Insbesondere sind auch Regeln einbezogen, die n^2 Zähler benutzen, um das Übergangsverhalten der Zugriffe durch relative Häufigkeiten anzunähern. Obwohl die Nach-Vorne-Schieben-Regel kein Gedächtnis benutzt, ist ihre erwartete Suchzeit für gewisse Klassen abhängiger Zugriffe optimal.

In Kapitel 4 werden wir weitere Regeln für Listen bezüglich ihrer erwarteten Suchzeit vergleichen.

In [9] wird die Menge $\mathcal{F}^*$ der Übergangsmatrizen von Markov-Ketten definiert, für die die Nach-Vorne-Schieben-Regel optimale asymptotische Kosten erzeugt. Es wird gezeigt, daß diese Menge eine gewisse Reichhaltigkeit besitzt:

Satz 1.26 (Chassaing 1993)
Sei P in $\mathcal{F}^*$ und $L = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ mit $0 \leq \lambda_i \leq 1$. Dann liegt auch die Matrix P_L in $\mathcal{F}^*$:

$$P_L = L \cdot P + (I - L).$$

Die Matrix P_L entsteht aus P, indem die Trägheit der Übergänge erhöht wird (elementweise Lokalität).

Folgende Elemente der Klasse $\mathcal{F}^*$ werden explizit angegeben:

Satz 1.27 (Chassaing 1993)

(a) Sei der Zustandsraum partitioniert in w Klassen der gleichen Größe k: $S = X_1 \cup \cdots \cup X_w$. Dann ist P definiert als

$$p_{ij} = \left\{ \begin{array}{lcl} \alpha & : & \textit{falls } i \textit{ und } j \textit{ in der gleichen Klasse liegen} \\ \beta & : & \textit{sonst} \end{array} \right\}$$

Im Falle $\alpha \geq \beta$ (Lokalität) gilt: $P \in \mathcal{F}^*$.

(b) Die Tridiagonalmatrix P mit $\alpha + \beta + \gamma = 1$,

$$P = \begin{pmatrix} \alpha & \beta & & & \gamma \\ \gamma & \alpha & \beta & & \\ & \gamma & \alpha & \beta & \\ & & & \ddots & \\ \beta & & & \gamma & \alpha \end{pmatrix}$$

liegt in $\mathcal{F}^*$, falls gilt

$$\alpha \geq (\beta - \gamma)(\beta^{n-1} + \gamma^{n-1})/(\beta^{n-1} - \gamma^{n-1}).$$

Wenn die Übergangsmatrix der Zugriffsfolge in der Klasse $\mathcal{F}^*$ liegt, dann erreicht die Nach-Vorne-Schieben-Regel die optimale Suchzeit bei sequentieller Suche. Hierbei geschieht die Umordnung der Liste in jedem Schritt.
Durch Verwendung von n^2 Zählern für die relativen Übergangshäufigkeiten kann man die Markov-Kette beliebig genau annähern. Bei einem Zustandswechsel muß aber die gesamte Liste umgeordnet werden, um die optimale Sortierung für die nächste Suche sicherzustellen. Bei diesem Umordnungsprozeß müssen die zu bewegenden Elemente auch gesucht werden. Dieser Aufwand gehört nicht zur eigentlichen Suche, muß aber dennoch hinzugezählt werden, wenn man den gleichen Optimalitätsbegriff zugrundelegt, der bei der Nach-Vorne-Schieben-Regel angewendet wird [10].

Wir erläutern dies an einem Beispiel.
Die Übergangsmatrix

$$P = \begin{pmatrix} \alpha & \alpha & \beta & \beta \\ \alpha & \alpha & \beta & \beta \\ \beta & \beta & \alpha & \alpha \\ \beta & \beta & \alpha & \alpha \end{pmatrix}$$

gehört gemäß Satz 1.27 zur Klasse $\mathcal{F}^*$, wenn man die Lokalität der Zugriffe fordert: $\alpha \geq \beta$, wobei $\beta = (1 - 2\alpha)/2$.
Die erwarteten Suchkosten unter der Nach-Vorne-Schieben-Regel errechnen sich mit

Formel (1.24) zu

$$\mu(\textsc{mtf}) = 3/2 + 4 \cdot \frac{2\alpha - 1}{4\alpha - 3}.$$

Die optimale Strategie untersucht erst die Elemente mit höherer Wahrscheinlichkeit α, und dann die Elemente mit Zugriffswahrscheinlichkeit β. Da die stationäre Verteilung die Gleichverteilung ist, gilt

$$\mu(\textsc{opt}) = (1 + 2)\alpha + (3 + 4)\beta = 7/2 - 4\alpha.$$

Der zulässige Bereich für α, in dem Lokalität herrscht, ist $1/4 < \alpha < 1/2$. In diesem Bereich gilt aber:

$$\mu(\textsc{opt}) < \mu(\textsc{mtf}).$$

Die Differenz kommt dadurch zustande, daß bei obiger Berechnung von $\mu(\textsc{opt})$ keine sequentielle Suche, sondern wahlfreier Zugriff auf die Elemente vorausgesetzt wird. Die Umordnungskosten, die beim Wechsel zwischen den Zustandsmengen $\{1, 2\}$ und $\{3, 4\}$ auftreten, sind für den Unterschied verantwortlich.

Kapitel 2

Binäre Bäume

2.1 Problemstellung

Wir betrachten wieder das statische Wörterbuchproblem. Zusätzlich zu den Voraussetzungen des letzten Kapitels wird gefordert, daß auf der zu speichernden Menge eine lineare Ordnung existiert, so daß man vergleichsorientiert suchen kann.

Sei also die Menge

$$S = \{1, 2, \ldots, n\}$$

zusammen mit der üblichen Ordnung auf den natürlichen Zahlen gegeben. Die Elemente werden in einem knotenorientierten binären Baum gespeichert. Jeder Knoten des Baums enthält ein Element x aus S, einen Verweis auf den linken Unterbaum und einen Verweis auf den rechten Unterbaum. Der linke Unterbaum enthält nur Elemente y mit $y < x$, der rechte nur Elemente y mit $y > x$.
Die Suche nach einem Element beginnt bei der Wurzel. An jedem Knoten wird das aktuelle Element mit dem zu suchenden verglichen. Wenn der Vergleich noch nicht erfolgreich war, wird durch einen weiteren Vergleich festgestellt, in welchem Unterbaum die Suche fortgeführt wird.
Die Kosten der Suche sind gleich der Anzahl der besuchten Knoten (= Tiefe des Elements im Baum). Wir untersuchen wieder verschiedene Zugriffsfolgen und die dazugehörigen erwarteten Kosten für die Suche.

Bei obigem Modell gehen wir davon aus, daß der Baum explizit mit Zeigern dargestellt ist. In einem sortierten Array erreicht man logarithmische Suchzeit durch Binärsuche. Wenn man an dem Array festhalten möchte, aber komplizierte Operationen erlaubt, kommt man zu sogenannten *impliziten* Datenstrukturen. Selbst-Organisations-Heuristiken für implizite Datenstrukturen wurden von Frederickson [15] betrachtet.

2.2 Unabhängige Zugriffe

Sei eine Wahrscheinlichkeitsverteilung

$$\vec{p} = (p_1, p_2, \ldots, p_n), \qquad p_i > 0, \qquad \sum_{i=1}^{n} p_i = 1$$

auf S gegeben. Die Zugriffe sind unabhängig und identisch verteilt mit der Verteilung $\vec{p}$.

2.2.1 Optimale Strategien

Wenn die Wahrscheinlichkeitsverteilung $\vec{p}$ bekannt ist, dann kann man einen statischen Suchbaum konstruieren, der minimale erwartete Suchkosten (minimale gewichtete Pfadlänge) aufweist. Dieser Suchbaum kann in Zeit $O(n^2)$ gebaut werden, wenn man dynamische Programmierung benutzt [29].

Sei μ_{opt} die erwartete Suchzeit eines optimalen statischen Suchbaums, und sei

$$H = H(\vec{p}) = \sum_{i=1}^{n} -p_i \log p_i$$

die **Entropie** der Verteilung $\vec{p}$. Dabei gelte $0 \log 0 := 0$. Hier und im folgenden ist log der Logarithmus zur Basis 2.

Lemma 2.1 (Bayer 1975)
Für die minimale erwartete Suchzeit μ_{opt} *gelten für alle* $d \in \mathbb{R}$ *die Abschätzungen*

$$(H - d)/(\log(2 + 2^{-d})) \leq \mu_{\mathrm{opt}} \leq H + 1$$

Speziell folgen die unteren Schranken

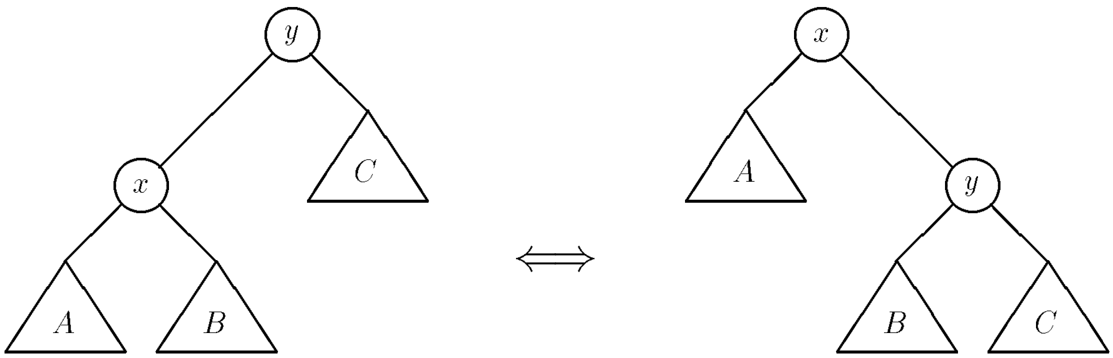
$$\begin{array}{rcl} d = 0 & : & \mu_{\mathrm{opt}} \geq H/\log 3 \\ d = \log(\mu_{\mathrm{opt}}) - 1 & : & \mu_{\mathrm{opt}} + \log \mu_{\mathrm{opt}} + \log e - 1 \geq H \end{array}$$

■

2.2.2 Die Zur-Wurzel-Rotieren-Regel

Zwei Heuristiken ohne Gedächtnis wurden 1978 von Allen und Munro vorgeschlagen und analysiert [2].

Die Einfacher-Austausch-Regel (*simple exchange*) bringt das nachgefragte Element um eine Position näher zur Wurzel, wenn es sich nicht schon an der Wurzel befindet. Das geschieht durch eine Rotation, um die Ordnung auf dem Baum zu erhalten:



Diese Regel entspricht der Vertauschen-Heuristik bei Listen, bei der das Element um eine Position näher zum Listenkopf bewegt wird. Wenn die Zugriffe gleichverteilt sind, dann ist die erwartete Suchzeit allerdings proportional zu $\sqrt{\pi n}$ (siehe [2]). Deshalb betrachten wir ausschließlich die Zur-Wurzel-Rotieren-Regel.

Die Zur-Wurzel-Rotieren-Regel (*move-to-root*) bringt das nachgefragte Element durch eine Folge von Rotationen zur Wurzel des Suchbaums. Sie entspricht der Nach-Vorne-Schieben-Regel bei den Listen. Der Aufwand für die Umordnung gemäß dieser Regel ist proportional zu den Suchkosten für das nachgefragte Element.

Zur Analyse machen wir die folgende Beobachtung:

Lemma 2.2 (Strukturlemma für die Zur-Wurzel-Rotieren-Regel)
Sei $i < j$. Das Element j ist ein Vorfahre von i genau dann, wenn der letzte Zugriff auf j nach dem letzten Zugriff auf eines der Elemente $i, \ldots, j-1$ erfolgt ist. Umgekehrt ist i ein Vorfahre von j genau dann, wenn der letzte Zugriff auf i nach dem letzten Zugriff auf eines der Elemente $i+1, \ldots, j$ aufgetreten ist.

Beweis:
Sei $i < j$. Aufgrund der Zur-Wurzel-Rotieren-Regel wird ein Element i Vorfahre von einem Element j höchstens dann, wenn i nachgefragt wurde. Wenn i nachgefragt wird, wird es zur Wurzel des Baums und damit Vorgänger aller Elemente. Dabei verliert der

Knoten j alle Vorgänger k mit $k \leq i$. Das erkennt man an obigem Bild. Sei x der Knoten mit Inhalt i, und y der Knoten mit Inhalt j. Durch Zugriff auf i geht man vom linken Baum zum rechten Baum über. Der Knoten y verliert alle Nachfolger im Unterbaums A und i. Das sind aber alle Elemente $k \leq i$. Wenn also nach dem letzten Zugriff auf j kein Zugriff mehr auf $i, \ldots, j-1$ erfolgt ist, dann ist j noch immer Vorfahre von i. Für $i > j$ ist die Argumentation analog. ■

Satz 2.3 (Allen/Munro 1978)
Bei unabhängigen, identisch verteilten Zugriffen gilt für die erwartete Suchzeit μ bei Verwendung der Zur-Wurzel-Rotieren-Regel:

$$\mu = 1 + 2 \sum_{i<j} \frac{p_i p_j}{p_i + \cdots + p_j} \tag{2.1}$$

Beweis:
Seien für die Zur-Wurzel-Rotieren-Regel definiert:

$d(j,i)$: asymptotische Wahrscheinlichkeit, daß j Vorfahre von i ist
$\mu(i)$: mittlere Tiefe von Element i

Sei $i < j$. Die Wahrscheinlichkeit, daß j Vorfahre von i ist, ergibt sich gemäß dem Strukturlemma daraus, daß für beliebige $k \geq 0$ die letzen $k+1$ Zugriffe aus einem Zugriff auf j gefolgt von k Nachfragen für Elemente verschieden von $i, \ldots, j$ bestanden. Also ist

$$\begin{aligned} d(j,i) &= p_j \cdot \sum_{k \geq 0} (1 - (p_i + \cdots + p_j))^k \\ &= \frac{p_j}{p_i + \cdots + p_j} \end{aligned}$$

Analog ergibt sich die Wahrscheinlichkeit, daß i ein Vorfahre von j ist, zu

$$d(i,j) = \frac{p_i}{p_i + \cdots + p_j}$$

Die erwartete Tiefe von Element i ist damit

$$\mu(i) = 1 + \sum_{j \neq i} d(j,i)$$

und die erwartete Suchzeit bei der Zur-Wurzel-Rotieren-Regel

$$\begin{aligned} \mu &= \sum_{i=1}^{n} p_i \cdot \mu(i) \\ &= 1 + \sum_{i=1}^{n} \sum_{j \neq i} \frac{p_i p_j}{p_i + \cdots + p_j} \\ &= 1 + 2 \sum_{i<j} \frac{p_i p_j}{p_i + \cdots + p_j} \end{aligned}$$

■

Dieses Ergebnis kann mit der Entropie H der Verteilung verglichen werden.

Satz 2.4 (Allen/Munro 1978)
Für die erwartete Suchzeit bei der Zur-Wurzel-Rotieren-Regel gilt

$$\mu \leq 1 + 2\ln 2 \cdot H$$

Beweis:
Definiere $s_{i,j} = (p_i + \cdots + p_j)/p_i$ für $i \leq j$. Dann ist $1 = s_{i,i} \leq s_{i,i+1} \leq \cdots \leq s_{i,n} \leq 1/p_i$ und

$$\begin{aligned}
\mu &= 1 + 2\sum_{i=1}^{n-1} p_i \sum_{j=i+1}^{n} \frac{s_{i,j} - s_{i,j-1}}{s_{i,j}} \\
&\leq 1 + 2\sum_{i=1}^{n-1} p_i \int_{s_{i,i}}^{s_{i,n}} \frac{1}{t} dt \\
&= 1 + 2\sum_{i=1}^{n-1} p_i \ln s_{i,n} \\
&\leq 1 + 2\sum_{i=1}^{n-1} p_i(-\ln p_i) \\
&= 1 + 2\ln 2 \cdot H
\end{aligned}$$

■

Satz 2.5 (Allen/Munro 1978)
Die erwartete Suchzeit läßt sich mit der optimalen Suchzeit vergleichen:

$$\begin{aligned}
\mu &\leq 1 + 2\ln 2 \cdot \Big(\mu_{\text{opt}} + \log \mu_{\text{opt}} + \log e - 1\Big) && (2.2) \\
&\leq 1.6138 + 1.3863 \cdot \Big(\mu_{\text{opt}} + \log \mu_{\text{opt}}\Big) && (2.3)
\end{aligned}$$

Beweis:
Mit Hilfe von Lemma 2.1 und der unteren Schranke bei $d = \log(\mu_{\text{opt}}/2)$ folgt sofort die Abschätzung

$$\begin{aligned}
\mu &\leq 1 + 2\ln 2 \cdot H \\
&\leq 1 + 2\ln 2 \cdot \Big(\mu_{\text{opt}} + \log \mu_{\text{opt}} + \log e - 1\Big)
\end{aligned}$$

■

Experimente zur empirischen Bestimmung der Suchzeiten bei verschiedenen Heuristiken für selbst-organisierende binäre Suchbäume wurden von Bell/Gupta [5] durchgeführt.

2.3 Abhängige Zugriffe

Wir betrachten das Verhalten der Zur-Wurzel-Rotieren-Regel, wenn die Folge der Zugriffe durch eine ergodische Markov-Kette beschrieben werden kann. Zuerst stellt sich die Frage, wie gut solche Folgen von Suchanfragen überhaupt bearbeitet werden können.

2.3.1 Optimale Strategien

Ein Vergleich mit dem Optimum ist wie bei unabhängigen Zugriffen über die Entropie möglich. Für eine ergodische Markov-Kette mit Übergangsmatrix $P = (p_{ij})$ und stationärer Verteilung (q_i) ist die **Entropie der Markov-Kette** definiert als

$$H(P) = \sum_{i=1}^{n} q_i \cdot H_i,$$

wobei

$$H_i = -\sum_{j=1}^{n} p_{ij} \cdot \log p_{ij}$$

die Entropie der Verteilung in Zustand i (Zeile i der Übergangsmatrix) ist. Dies ist eine gute Abschätzung für die optimale erwartete Suchzeit wegen folgendem Kodierungssatz.

Satz 2.6:

Man kann eine präfixfreie binäre Kodierung einer Markov-Kette P konstruktiv angeben, deren erwartete Codewort-Länge L optimal ist mit

$$H(P) + 1 \leq L \leq H(P) + 2.$$

■

Man kann eine Entsprechung zwischen binären Kodierungen und binären Suchbäumen herstellen (siehe Ahlswede/Wegener [1]). Für den Fall abhängiger Zugriffe müssen die Suchbäume für jeden Zustand zu einem Suchgraphen vereinigt werden. Wenn man die Kette P der Anfragen kennt, dann kann man einen Suchgraphen bauen, der optimale erwartete Suchzeit aufweist.

Satz 2.7 (Hotz 1993)

Man kann einen optimalen Suchgraphen konstruieren, so daß für die erwartete Suchzeit μ_{opt} in diesem Graphen gilt

$$H(P) \leq \mu_{\mathrm{opt}} \leq H(P) + 2 \tag{2.4}$$

Die Konstruktion ist in [20] dargestellt. Der Suchgraph braucht wie die Übergangsmatrix $\Theta(n^2)$ Platz. Unter geeigneten Annahmen über die Lokalität der Zugriffe kann man die Größe des Graphen reduzieren, wobei die erwartete Suchzeit nur gering erhöht wird. Genauer gilt

Satz 2.8 (Hotz 1993)
Der Zustandsraum sei in k Teilmengen $S = X_1 \cup \ldots \cup X_k$ partitioniert, so daß Übergänge zwischen verschiedenen Teilmengen mit Wahrscheinlichkeit $\leq \varepsilon/n$ auftreten. Dann kann man den Suchgraphen zerlegen, so daß der resultierende Graph noch die Größe

$$\sum_{i=1}^{k} |X_i|^2 \; < \; n^2$$

aufweist. Die erwartete Suchzeit ist dann

$$H(P) + O(\varepsilon) \leq \mu \leq H(P) + 2 + O(\varepsilon).$$

Wir stellen uns die Frage, wie gut die erwartete Suchzeit noch sein kann, wenn man nur linear viel Speicherplatz zuläßt und die Trägheit der Übergänge voraussetzt. Dies führt zur Betrachtung von selbst-organisierenden Suchbäumen bei abhängigen Zugriffen.

2.3.2 Die Zur-Wurzel-Rotieren-Regel

Zur Analyse der Zur-Wurzel-Rotieren-Regel bei abhängigen Zugriffen formen wir das Strukturlemma 2.2 um, damit eine Rückwärtsanalyse möglich wird. Dies geschieht analog zu der Umformulierung des Strukturlemmas für die Nach-Vorne-Schieben-Regel (vgl. Lemma 1.12).

Lemma 2.9:

Sei $i < j$. Wenn auf k zugegriffen wird, dann ist j ein Vorfahre von i genau dann, wenn wir die Zugriffsfolge bei k startend zurückverfolgen und auf j treffen, bevor eines der Elemente $i, \ldots, j-1$ aufgetreten ist. Entsprechend ist i ein Vorfahre von j beim Zugriff auf k genau dann, wenn wir die Zugriffsfolge bei k startend zurückverfolgen und auf i treffen, bevor eines der Elemente $i+1, \ldots, j$ aufgetreten ist.

Sei $d(j, i|k)$ die asymptotische Wahrscheinlichkeit, daß j Vorfahre von i ist, wenn gerade auf k zugegriffen wird. Wir benötigen die erwartete Suchzeit für das Element i, wenn auf i zugegriffen wird:

$$\mu(i|i) = 1 + \sum_{j \neq i} d(j, i|i).$$

Sei $i < j$. Gemäß der Umformulierung 2.9 des Strukturlemmas betrachten wir eine rückwärtslaufende Kette mit der Tabu-Menge $H = \{i, \ldots, j-1\}$, um $d(j, i|i)$ zu berechnen. Wie bei linearen Listen muß die Berechnung für alle Werte von i und j mit $i \neq j$ durchgeführt werden. Allerdings ist die Größe der Tabu-Mengen nicht mehr konstant 1, sondern $|j - i|$.

Prinzipiell kann man die gesuchten Werte mit der Formel von Satz 1.10 berechnen:

$$d(j, i|i) = {}_H\hat{f}^*_{ij} = \frac{\det((\delta_{ab} - 1)\hat{m}_{ab} + \hat{m}_{ib} + \hat{m}_{aj} - \hat{m}_{ij})_{a,b \in H}}{\det((\delta_{ab} - 1)\hat{m}_{ab} + \hat{m}_{jb} + \hat{m}_{aj})_{a,b \in H}}$$

Leider läßt sich die Berechnung der auftretenden Determinanten auch für spezielle Typen von Markov-Ketten nicht mehr allgemein handhaben. Deshalb benutzen wir im folgenden eine elementare Berechnungsmethode, die zumindest für Spezialfälle zu Ergebnissen führt. Außerdem sind die auftretenden Größen wieder leicht interpretierbar.

Wie in Kapitel 1 führt der Ansatz über absorbierende Markov-Ketten. Alle Tabu-Zustände werden absorbierend, und zusätzlich der Zustand j. Wir betrachten die Wahrscheinlichkeit der Absorption in j. Da der Startzustand i auch tabu ist, müssen wir beim ersten Schritt die alte Situation der Zeile j betrachten; nachdem j dann verlassen ist, wird er absorbierend.

Satz 2.10:

> *Für $i < j$ gilt*
>
> $$d(j, i \mid i) = \hat{p}_{ij} + \sum_{k \in T} \sum_{k' \in T} \hat{p}_{ik} \hat{n}_{kk'} \hat{p}_{k'j}, \qquad (2.5)$$
>
> *wenn $T = \{1, \ldots, i-1\} \cup \{j+1, \ldots, n\}$ die Menge der transienten Zustände ist, $\hat{Q} = \hat{P}|_{T \times T}$ die Übergangsmatrix der transienten Zustände in der rückwärtslaufenden Kette und $\hat{N} = (I - \hat{Q})^{-1}$ die Fundamentalmatrix der rückwärtslaufenden, absorbierenden Kette.*

Für $i > j$ gilt die Aussage analog. Der Satz ist eine direkte Übertragung von Satz 1.16 auf Seite 19, mit entsprechender Interpretation der Tabu-Menge H.

Im Spezialfall unabhängiger Zugriffe erhält man wieder die Ergebnisse aus Abschnitt 1, Satz 2.3:

Korollar 2.11 :

Für unabhängige Zugriffe, d.h. $p_{ij} = p_j$ für alle $i, j = 1, \ldots, n$, gilt für alle $i < j$:

$$\begin{aligned} d(j, i|i) &= \frac{p_j}{p_i + \cdots + p_j} \\ d(i, j|j) &= \frac{p_i}{p_i + \cdots + p_j} \end{aligned}$$

Beweis:
Sei die Übergangsmatrix P für unabhängige Zugriffe gegeben als

$$P = \begin{pmatrix} p_1 & p_2 & \cdots & p_n \\ p_1 & p_2 & \cdots & p_n \\ \vdots & & & \vdots \\ p_1 & p_2 & \cdots & p_n \end{pmatrix}$$

Da die Zugriffe unabhängig sind, ist die rückwärtslaufende Kette gleich der vorwärtslaufenden.
Betrachte für beliebige $i < j$ die Tabu-Menge $H = \{i, \ldots, j\}$ und ihr Komplement $T = S - H$, die Menge der transienten Zustände. Sei $Q = P|_{T \times T}$ die Übergangsmatrix der transienten Zustände mit Zeilensumme $s = \sum_{k \in T} p_k = 1 - (p_i + \cdots + p_j)$ in jeder Zeile. Dann ist $Q^2 = s \cdot Q$.
Die Fundamentalmatrix N der absorbierenden Kette ist

$$\begin{aligned} N &= (I - Q)^{-1} \\ &= I + \sum_{m \geq 1} Q^m \\ &= I + Q \cdot \sum_{m \geq 0} s^m \\ &= I + \frac{1}{1-s} \cdot Q \end{aligned}$$

Damit ist

$$\begin{aligned} d(j, i|i) &= p_{ij} + \sum_{k \in T} \sum_{k' \in T} p_{ik} n_{kk'} p_{k'j} \\ &= p_j + \sum_{k \in T} \sum_{k' \in T} p_k n_{kk'} p_j \\ &= p_j + p_j \cdot \sum_{k \in T} p_k \cdot \sum_{k' \in T} n_{kk'} \\ &= p_j + p_j \cdot \sum_{k \in T} p_k \cdot \left(1 + \frac{s}{1-s}\right) \end{aligned}$$

$$
\begin{aligned}
&= p_j + p_j \cdot s \cdot \left(1 + \frac{s}{1-s}\right) \\
&= p_j \cdot \left(1 + s + \frac{s^2}{1-s}\right) \\
&= \frac{p_j}{1-s} \\
&= \frac{p_j}{p_i + \cdots + p_j}
\end{aligned}
$$

Die Berechnung für $d(i, j|j)$ verläuft analog. ■

2.3.3 Beispiel

In diesem Abschnitt wird die erwartete Suchzeit unter der Zur-Wurzel-Rotieren-Regel für gewisse Klassen von Zugriffsfolgen konkret berechnet.

Die Markov-Kette der Nachfragen soll wie im Beispiel 2 für Listen (siehe Seite 22) jede vorgegebene stationäre Verteilung annehmen können, und durch einen Parameter λ soll die elementweise Lokalität regulierbar sein.

Sei $\vec{p} = (p_1, \ldots, p_n)$ eine Wahrscheinlichkeitsverteilung und $0 < \lambda \leq 1$. Weiter sei A die Übergangsmatrix unabhängiger Zugriffe, bei der alle Zeilen $\vec{p}$ sind. Betrachte

$$P = \lambda A \; + \; (1-\lambda) I. \tag{2.6}$$

Wie beim Beispiel für die linearen Listen gesehen, ist die stationäre Verteilung $\vec{q} = \vec{p}$.

Die Kette ist zeit-umkehrbar, denn es gilt:

$$
\begin{aligned}
\hat{p}_{ij} &= \frac{q_j}{q_i} p_{ji} \\
&= \frac{q_j}{q_i} (\lambda q_i + (1-\lambda)\delta_{ji}) \\
&= \lambda q_j + (1-\lambda)\delta_{ij} \\
&= p_{ij}
\end{aligned}
$$

Zur Berechnung von $d(j, i|i)$ betrachten wir die Tabu-Menge $H = \{i, \ldots, j\}$, mit $i < j$ fest. Die Menge der transienten Zustände ist dann $T = S - H$. Sei $Q = P|_{T \times T}$ die Übergangsmatrix der transienten Zustände. Wenn man $R = A|_{T \times T}$ setzt, dann ist

$$Q = \lambda R + (1-\lambda) I.$$

Die Zeilensumme von R ist $s = 1 - (q_i + \cdots + q_j)$, und es gilt $R^2 = s \cdot R$. Wir fahren fort wie im Beweis zu Lemma 1.19. Für Q^m gilt die Rekursionsgleichung

$$Q^m = a^{(m)} \cdot R + b^{(m)} \cdot I$$

mit

$$a^{(1)} = \lambda, \qquad b^{(1)} = 1 - \lambda$$

sowie

$$\begin{aligned} Q^{m+1} &= Q \cdot Q^m \\ &= \underbrace{\left(\lambda s a^{(m)} + (1-\lambda) a^{(m)} + \lambda b^{(m)}\right)}_{a^{(m+1)}} \cdot R \ + \ \underbrace{(1-\lambda) b^{(m)}}_{b^{(m+1)}} \end{aligned}$$

Für die Koeffizienten folgt

$$\begin{aligned} b^{(m+1)} &= (1-\lambda)^{m+1} \\ a^{(m+1)} &= (\lambda s + 1 - \lambda) a^{(m)} + \lambda (1-\lambda)^m \\ &= (\lambda s + 1 - \lambda)^m a^{(1)} + \lambda \sum_{j=0}^{m-1} (\lambda s + 1 - \lambda)^j (1-\lambda)^{m-j} \\ &= \lambda \sum_{j=0}^{m} (\lambda s + 1 - \lambda)^j (1-\lambda)^{m-j} \\ &= \lambda (1-\lambda)^m \sum_{j=0}^{m} \left(1 + \frac{\lambda s}{1-\lambda}\right)^j \\ &= \left((\lambda s + 1 - \lambda)^{m+1} - (1-\lambda)^{m+1}\right)/s \end{aligned}$$

Für die Fundamentalmatrix der transienten Zustände folgt

$$\begin{aligned} N &= I + \sum_{m=1}^{\infty} Q^m \\ &= I + \frac{1}{s} R \cdot \sum_{m=1}^{\infty} (\lambda s + 1 - \lambda)^m + (I - \frac{1}{s} R) \cdot \sum_{m=1}^{\infty} (1-\lambda)^m \\ &= I + R \, \frac{\lambda s + 1 - \lambda}{\lambda s (1-s)} \ + \ (I - \frac{1}{s} R) \, \frac{1-\lambda}{\lambda} \end{aligned}$$

und für die Zeilensumme von N in jeder Zeile k:

$$\sum_{k' \in T} n_{kk'} = 1 + \frac{\lambda s + 1 - \lambda}{\lambda (1-s)} = \frac{1}{\lambda (1-s)}.$$

Zusammen gilt für die asymptotische Wahrscheinlichkeit, daß j vor i steht beim Zugriff auf i:

$$d(j, i | i) = p_{ij} + \sum_{k \in T} \sum_{k' \in T} p_{ik} n_{kk'} p_{k'j}$$

$$
\begin{aligned}
&= \lambda q_j + \sum_{k\in T}\sum_{k'\in T} \lambda q_k \cdot n_{kk'} \cdot \lambda q_j \\
&= \lambda q_j \cdot \left(1 + \frac{1}{\lambda(1-s)} \sum_{k\in T} \lambda q_k\right) \\
&= \lambda q_j \cdot \left(1 + \frac{\lambda s}{\lambda(1-s)}\right) \\
&= \lambda q_j \frac{1}{1-s} \\
&= \frac{\lambda q_j}{q_i + \cdots + q_j}
\end{aligned}
$$

Für $d(i,j|j)$ verläuft die Berechnung analog.

Satz 2.12:

Sei $\mu(A)$ die erwartete Suchzeit, die die Zur-Wurzel-Rotieren-Regel bei unabhängigen Zugriffen mit der Verteilung $\vec{q}$ produziert (vgl. Formel (2.1)). Dann gilt für die erwartete Suchzeit unter der Markov-Kette P:

$$\mu(P) = \lambda \cdot \mu(A) + (1-\lambda) \tag{2.7}$$

Beweis:
Die erwartete Suchzeit ist

$$
\begin{aligned}
\mu &= \sum_i q_i \cdot \mu(i|i) \\
&= \sum_i q_i \cdot \left(1 + \sum_{j\neq i} d(j,i|i)\right) \\
&= 1 + \lambda \sum_i \sum_{j\neq i} \frac{q_i q_j}{q_i + \cdots + q_j} \qquad (2.8) \\
&= \lambda \cdot \mu(A) + (1-\lambda)
\end{aligned}
$$

■

Die Analogie von Formel (2.7) für Bäume und Formel (1.18) für Listen ist bemerkenswert. Wie in Satz 1.21 können wir auch für Bäume zeigen, daß die erwartete Suchzeit beliebig nahe an 1 kommen kann.

Satz 2.13:

Für eine beliebige Wahrscheinlichkeitsverteilung $\vec{p} = (p_1, \ldots, p_n)$, $p_i > 0$, und jedes $\varepsilon > 0$ existiert eine Markov-Kette P, so daß $\vec{p}$ die stationäre Verteilung der Kette ist und für die erwartete Suchzeit μ unter der Zur-Wurzel-Rotieren-Regel gilt

$$\mu < 1 + \varepsilon \tag{2.9}$$

■

Wir wollen die erwartete Suchzeit mit der Entropie der Markov-Kette vergleichen. Das Gruppierungs-Axiom für die Entropie lautet für eine Wahrscheinlichkeitsverteilung $(p_0, p_1, \ldots, p_n)$, $\sum_{j=0}^{n} p_j = 1$, $i = 1, \ldots, n$:

$$H(p_0, \ldots, p_n) = H(p_1, \ldots, p_0 + p_i, \ldots, p_n) + (p_0 + p_i) \cdot H\left(\frac{p_0}{p_0 + p_i}, \frac{p_i}{p_0 + p_i}\right) \tag{2.10}$$

Außerdem gilt die Abschätzung

$$0 \leq H(p_0, \ldots, p_n) \leq \log(n + 1) \tag{2.11}$$

Wir setzen $p_j = \lambda q_j$ $(j = 1, \ldots, n)$ und $p_0 = 1 - \lambda$, und betrachten die i-te Zeile von P. Für die Entropie H_i dieser Zeile gilt

$$\begin{aligned}
H_i &= H(p_1, \ldots, p_i + p_0, \ldots, p_n) \\
&= H(p_0, \ldots, p_n) - (p_0 + p_i) \cdot H\left(\frac{p_0}{p_0 + p_i}, \frac{p_i}{p_0 + p_i}\right) \\
&= -(1 - \lambda)\log(1 - \lambda) - \sum_{j=1}^{n} (\lambda q_j) \log(\lambda q_j) - (p_0 + p_i) \cdot H\left(\frac{p_0}{p_0 + p_i}, \frac{p_i}{p_0 + p_i}\right) \\
&\geq -(1 - \lambda)\log(1 - \lambda) - \lambda \log \lambda + \lambda H(q_1, \ldots, q_n) - (p_0 + p_i) \cdot \log 2 \\
&= H(\lambda, 1 - \lambda) + \lambda H(q_1, \ldots, q_n) - (1 - \lambda + \lambda q_i)
\end{aligned}$$

Dann gilt für die Entropie $H(P)$ der Markov-Kette

$$\begin{aligned}
H(P) &= \sum_{i=1}^{n} q_i H_i \\
&\geq \lambda H(\vec{q}) + H(\lambda, 1 - \lambda) - \sum_i q_i (1 - \lambda + \lambda q_i) \\
&= \lambda H(\vec{q}) + H(\lambda, 1 - \lambda) - (1 - \lambda) - \sum_i \lambda q_i^2 \\
&\geq \lambda H(\vec{q}) + H(\lambda, 1 - \lambda) - (1 - \lambda) - \lambda \\
&= \lambda H(\vec{q}) + H(\lambda, 1 - \lambda) - 1
\end{aligned}$$

Aus dem Fall unabhängiger Zugriffe kennen wir die Abschätzung $\mu(A) \leq 1+2\ln 2 \cdot H(\vec{q})$. Damit folgt

$$\begin{aligned}
\mu(P) &= \lambda\mu(A) + (1-\lambda) \\
&\leq \lambda(1 + 2\ln 2 \cdot H(\vec{q})) + (1-\lambda) \\
&= 1 + 2\ln 2 \cdot \lambda H(\vec{q}) \\
&\leq 1 + 2\ln 2 \cdot \Big(H(P) + 1 - H(\lambda, 1-\lambda)\Big)
\end{aligned}$$

Satz 2.14:

Für die Zugriffsfolge (2.6) gilt für die erwartete Suchzeit unter der Zur-Wurzel-Rotieren-Regel

$$\begin{aligned}
\mu &\leq 1 + 2\ln 2 \cdot \Big(H(P) + 1 - H(\lambda, 1-\lambda)\Big) && (2.12) \\
&\leq 1 + 2\ln 2 \cdot (H(P) + 1) && (2.13)
\end{aligned}$$

Satz 2.15:

Wenn μ_{opt} die asymptotische Suchzeit in einem optimalen Suchgraphen bezeichnet, dann gilt für die Zugriffsfolge (2.6)

$$\begin{aligned}
\mu &\leq 1 + 2\ln 2 \cdot \Big(\mu_{\text{opt}} + \log \mu_{\text{opt}} + \log e\Big) \\
&\leq 3 + 1.3863 \cdot (\mu_{\text{opt}} + \log \mu_{\text{opt}})
\end{aligned}$$

Beweis:
Wenn im Zustand i nach j gesucht wird, so wird dafür der optimale Suchbaum für den Zustand i benutzt. Für die erwartete Suchzeit μ_i gilt also für beliebige $d_i \in \mathbb{R}$ die Abschätzung aus Lemma 2.1

$$H_i \leq \mu_i \cdot \log(2 + 2^{-d_i}) + d_i$$

Im Suchgraphen kann man nun das gefundene Blatt j mit der Wurzel des Suchbaums für den Zustand j identifizieren. Vor Beginn der Suchanfragen muß man sich auf einen Startzustand einigen bzw. eine Kodierung des Startzustands übertragen. Dieser Aufwand spielt aber asymptotisch keine Rolle. Also gilt für den den erwarteten Suchaufwand die untere Schranke

$$H(P) = \sum_i q_i H_i$$

$$
\begin{aligned}
&\le \sum_i q_i \cdot \Big(\mu_i \cdot \log(2+2^{-d}) + d\Big) \\
&= \mu_{\text{opt}} \cdot \log(2+2^{-d}) + d \\
&\le \mu_{\text{opt}} + \log \mu_{\text{opt}} + \log e - 1
\end{aligned}
$$

wenn man $d = \log(\mu_{\text{opt}}/2)$ setzt und $(1+1/x)^x \le e$ für $x > 0$ beachtet, und damit

$$
\begin{aligned}
\mu &\le 1 + 2\ln 2 \cdot \Big(H(P) + 1\Big) \\
&\le 1 + 2\ln 2 \cdot \Big(\mu_{\text{opt}} + \log \mu_{\text{opt}} + \log e\Big) \\
&\le 3 + 1.3863 \cdot (\mu_{\text{opt}} + \log \mu_{\text{opt}})
\end{aligned}
$$

■

Spezialfall:
Wir betrachten wieder den Fall

$$
p_{ij} = \left\{ \begin{array}{ccc} \alpha & : & i = j \\ \beta & : & i \ne j \end{array} \right\} \tag{2.14}
$$

Dann ist $\vec{q} = (1/n, \ldots, 1/n)$ und
$\lambda = (1-\alpha)n/(n-1)$, damit $q_i q_j/(q_i + \cdots + q_j) = 1/((j-i+1)n)$ und

$$
\begin{aligned}
\mu(A) &= 1 + 2\sum_{i<j} \frac{1}{(j-i+1)n} \\
&= 1 + \frac{2}{n} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{j-i+1} \\
&= 1 + \frac{2}{n} \sum_{j=2}^{n} \frac{n+1-j}{j} \\
&= 1 + \frac{2}{n}\Big((n+1)(H_n - 1) - (n-1)\Big) \\
&= 1 + \frac{2}{n}\Big((n+1)H_n - 2n\Big) \\
&= 2\,\frac{n+1}{n} H_n - 3,
\end{aligned}
$$

und damit (H_n: n-te harmonische Zahl)

$$
\mu(P) = 1 + 2\lambda\left(\frac{n+1}{n} H_n - 2\right).
$$

Es gilt $H_n \le 1 + \ln n$. Man sieht leicht, daß für $n \ge 6$ gilt

$$
\frac{n}{n-1}\left(\frac{n+1}{n}\Big(1 + \ln n\Big) - 2\right) \le \ln(n-1)
$$

Für die Entropie der Kette finden wir

$$\begin{aligned}
H(P) &= -\alpha \log \alpha - (n-1)\beta \log \beta \\
&= -\alpha \log \alpha - (1-\alpha) \log \frac{1-\alpha}{n-1} \\
&= -\alpha \log \alpha - (1-\alpha) \log(1-\alpha) + (1-\alpha) \log(n-1) \\
&= H(\alpha, 1-\alpha) + (1-\alpha)(\ln(n-1))/\ln 2 \\
&\geq (1-\alpha)(\ln(n-1))/\ln 2
\end{aligned}$$

Damit gilt für die Zugriffsfolge (2.14) und $n \geq 6$

$$\begin{aligned}
\mu(P) &= 1 + 2\lambda \left(\frac{n+1}{n} H_n - 2 \right) \\
&= 1 + 2(1-\alpha) \frac{n}{n-1} \left(\frac{n+1}{n} H_n - 2 \right) \\
&\leq 1 + 2(1-\alpha) \ln(n-1) \\
&\leq 1 + 2 \ln 2 \cdot H(P) \\
&\leq 1 + 2 \ln 2 \cdot \left(\mu_{\text{opt}} + \log \mu_{\text{opt}} + \log e \right) \\
&\leq 1.6138 + 1.3863 \cdot (\mu_{\text{opt}} + \log \mu_{\text{opt}})
\end{aligned}$$

In diesem Spezialfall kann also die oben gefundene Schranke um den Summand $2 \ln 2 \cdot (1 - H(\lambda, 1-\lambda))$ verbessert werden.

Kapitel 3

Die k-Mal-Hintereinander-Strategien

3.1 Definition

Als nächstes betrachten wir wieder lineare Listen und eine Klasse von Regeln, die etwas Speicher benutzen dürfen. Die *k-Mal-Hintereinander-Strategien* (*k in a row heuristics*) werden in Verbindung mit einer der in Kapitel 1 beschriebenen Regeln verwendet. Die Aktion dieser Regel wird nur ausgeführt, wenn auf das gleiche Element k Mal hintereinander zugegriffen wurde. Die Heuristik darf also ein Gedächtnis der Größe $O(\log k + \log n)$ benutzen, um den aktuellen Stand des Zählers zwischen 1 und k und das zuletzt nachgefragte Element zu speichern. Diese Klasse von Regeln wurde von Gonnet/Munro/Suwanda [16] analysiert. Es gibt sie in zwei Ausprägungen:

Definition 3.1:

> *Seien $k > 0$ und eine Selbst-Organisations-Regel R gegeben (z.B. R=Nach-Vorne-Schieben-Regel oder R=Vertauschen-Regel).*
>
> *Die* **Simple-k-Heuristik** *ruft die Aktion der Regel R nur dann auf, wenn die letzten k Zugriffe auf das gleiche Element gerichtet waren.*
>
> *Die* **Batched-k-Heuristik** *gruppiert die Zugriffe in Blöcke der Länge k und ruft die Aktion der Regel R nur dann auf, wenn alle Zugriffe eines Blocks das gleiche Element zum Ziel hatten.*

Die Einteilung in Blöcke bei der Batched-k-Heuristik geschieht nur zum Zweck der Umordnung. Da nicht nur k aufeinanderfolgende Zugriffe auf ein Element gerichtet sein müssen, sondern diese auch noch genau in den Blockgrenzen liegen müssen, wird die Aktion der Regel R noch seltener angewendet als bei der Simple-k-Heuristik.
Bei $k = 1$ erhält man die gewöhnliche Regel R.

3.2 Unabhängige Zugriffe

Wir betrachten die k-Mal-Hintereinander-Strategien in Verbindung mit der Nach-Vorne-Schieben-Regel. Schon für kleine Werte k ergeben sich gute Verbesserungen der Suchzeit, wenn die Zugriffe unabhängig sind.

Satz 3.2:

Sei M die erwartete Suchzeit bei optimaler Ordnung der Liste bei unabhängigen Zugriffen. Dann gilt für die Simple-k-Nach-Vorne-Schieben-Regel:

$$\begin{aligned}
k = 1 &: \mu \leq 1.57079\ldots \cdot M\\
k = 2 &: \mu \leq 1.36602\ldots \cdot M\\
k = 3 &: \mu \leq 1.27388\ldots \cdot M\\
k = 4 &: \mu \leq 1.22788\ldots \cdot M
\end{aligned}$$

Für die Batched-k-Nach-Vorne-Schieben-Regel gilt:

$$\mu \leq \left(\frac{\pi}{2k} \csc \frac{\pi}{2k}\right) \cdot M$$

$$\begin{aligned}
k = 1 &: \mu \leq 1.57079\ldots \cdot M\\
k = 2 &: \mu \leq 1.11072\ldots \cdot M\\
k = 3 &: \mu \leq 1.04719\ldots \cdot M
\end{aligned}$$

Stets gilt, daß für festes k die Batched-k-Heuristik bessere erwartete Suchzeiten aufweist als die Simple-k-Heuristik. Für $k \to \infty$ gilt für beide Regeln: $\mu \to M$.

Der Faktor bei $k = 1$ ist $\pi/2$ gemäß Satz 1.2. Für die Faktoren für die Simple-k-Regel bei $k > 1$ sind nur numerische Annäherungen bekannt. Sie stammen von Gonnet/Munro/Suwanda [16], ebenso wie der Vergleich der Heuristiken und die Grenzwert-Aussage. Die Formel für die Batched-k-Regel ist in Chung/Hajela/Seymour [12] angegeben und stellt eine Verbesserung gegenüber numerisch gefundenen Werten dar.

Die Eigenschaft, daß für steigende k fallende erwartete Suchzeiten entstehen, konnte auch für die Batched-k-Vertauschen-Regel gezeigt werden (Nelson [30]).

Die Herleitung bedient sich wieder der Theorie der Markov-Ketten. Bei der Simple-k-Heuristik ist der Zustandsraum die Menge $\{1, \ldots, k\} \times \mathcal{S}_n$ mit $kn!$ Elementen, bei der Batched-k-Heuristik ist es die Menge $\mathcal{S}_n$ der Permutationen, jedoch sind die Übergangswahrscheinlichkeiten anders definiert. Diese Markov-Ketten sind ergodisch.

Sei die Zugriffsverteilung gegeben durch den Vektor $\vec{p} = (p_1, \ldots, p_n)$. Außerdem sei $b_k(j,i)$ die asymptotische Wahrscheinlichkeit, daß sich das Element j vor dem Element i in der Liste befindet, wenn man die Simple-k-Nach-Vorne-Schieben-Regel anwendet, sowie $b'_k(j,i)$ die asymptotische Wahrscheinlichkeit, daß sich das j vor i befindet, wenn man die Batched-k-Nach-Vorne-Schieben-Regel benutzt. Dann gilt:

Lemma 3.3 (Gonnet/Munro/Suwanda 1981)
Für die Simple-k-Nach-Vorne-Schieben-Regel ist die asymptotische Wahrscheinlichkeit, daß j vor i steht,

$$b_k(j,i) = \frac{p_j^k \sum_{l=0}^{k-1} p_i^l}{p_j^k \sum_{l=0}^{k-1} p_i^l + p_i^k \sum_{l=0}^{k-1} p_j^l} \tag{3.1}$$

Für die Batched-k-Nach-Vorne-Schieben-Regel ist diese Wahrscheinlichkeit

$$b'_k(j,i) = \frac{p_j^k}{p_i^k + p_j^k} \tag{3.2}$$

Diese Formeln werden in [16] gezeigt (Lemma 2.1.1 und Kapitel 2.2), und mit ihrer Hilfe die erwartete Suchzeit für die jeweiligen Regeln berechnet.

Im folgenden werde ich einen neuen Ansatz beschreiben, der die Analyse von Markov-Zugriffsfolgen ermöglicht. Die Formeln für die asymptotischen Wahrscheinlichkeiten $b_k(j,i|i)$ und $b'_k(j,i|i)$, daß j vor i steht, wenn gerade auf i zugegriffen wird, werden hergeleitet. Daraus ergeben sich die obigen Formeln für unabhängige Zugriffe als einfaches Korollar.
Bei abhängigen Zugriffen mit Lokalität werden wir sehen, daß mit wachsendem k die Reaktion der Heuristiken immer weiter verzögert wird und die erwartete Suchzeit steigt. Außerdem gilt bei Lokalität, daß für festes k die Simple-k-Regel stets besser abschneidet als die Batched-k-Regel. Wir stellen also fest, daß die Hierarchie der Heuristiken bzgl. der erwarteten Suchzeiten bei abhängigen Zugriffen mit trägem Übergangsverhalten gerade umgekehrt ist als bei unabhängigen Zugriffen.

3.3 Expandierte Markov-Ketten

In diesem Abschnitt beschreiben wir, wie man aus einer gegebenen Markov-Kette neue Ketten gewinnen kann, indem man Zustände zu Tupeln zusammenfaßt. Diese Tupel bilden die Zustände der neuen Kette. Die Grundlagen der Darstellung stammen aus [22], Seite 140-144, wo das Verfahren am Beispiel von Zwei-Tupeln erläutert ist.

Sei also eine ergodische Markov-Kette mit Zustandsmenge S und Übergangsmatrix P und ein $k > 1$ gegeben. Wir wollen k-Tupel von Elementen aus S als Zustände der neuen Markov-Kette $\tilde{P}$ betrachten. Dabei sollen $k-1$ Übergänge der ursprünglichen Kette durch einen Zustand der neuen Kette ausgedrückt werden. Es dürfen also nur die Zustände auftreten, deren korrespondierende Folgen von Übergängen in der alten Kette möglich sind, d.h. eine Wahrscheinlichkeit > 0 haben. Genauergesagt ist ein Zustand $[i_1 i_2 \dots i_k]$ der neuen Kette nur dann erlaubt, wenn $p_{i_1 i_2} \cdot p_{i_2 i_3} \cdot \ldots \cdot p_{i_{k-1} i_k} > 0$ gilt. Diese Einschränkung ist nicht unmittelbar notwendig; wir brauchen sie jedoch, wenn wir ergodische Ketten betrachten. Denn bei einer ergodischen Kette muß jeder Zustand mit einer Wahrscheinlichkeit > 0 auftreten.

Die Zustandsmenge $\tilde{S}$ der expandierten Kette ist also

$$\tilde{S} = \{ \, [i_1 i_2 \dots i_k] \mid i_l \in S \text{ für } l = 1, \dots, k \text{ und } p_{i_1 i_2} \cdots p_{i_{k-1} i_k} > 0 \, \}$$

Einen Übergang von $[i_1 \dots i_k]$ nach $[j_1 \dots j_k]$ kann man sich dadurch vorstellen, daß ein Fenster der Länge k um eine Position nach rechts auf der ursprünglichen Kette verschoben wird. Die Wahrscheinlichkeit dieses Übergangs wird durch die Wahrscheinlichkeit gegeben, mit der ein Übergang vom letzten Element des alten Fensters zum letzten Element des nächsten Fensters erfolgt. Dabei werden alle anderen Elemente um eine Position nach links geschoben, und das erste Element tritt links aus dem Fenster aus. Also ist

$$\tilde{p}_{[i_1 i_2 \dots i_k][i_2 i_3 \dots i_{k+1}]} = p_{i_k i_{k+1}}$$

und

$$\tilde{p}_{[i_1 \dots i_k][j_1 \dots j_k]} = 0, \quad \text{falls es ein } l \text{ mit } i_{l+1} \neq j_l \text{ gibt, } 1 \leq l < k.$$

Mit dem Kronecker-Symbol δ läßt sich dies schreiben als

$$\tilde{p}_{[i_1 \dots i_k][j_1 \dots j_k]} = p_{i_k j_k} \cdot \delta_{[i_2 \dots i_k][j_1 \dots j_{k-1}]}$$

Die Markov-Kette mit Zustandsraum $\tilde{S}$ und Übergangsmatrix $\tilde{P}$ wie oben definiert heißt **expandierte Markov-Kette**.

Im folgenden lassen wir die Tilde bei den Wahrscheinlichkeiten der expandierten Kette wieder weg. Man erkennt sie daran, daß die Indizes k-Tupel sind.

Satz 3.4:

> *Die expandierte Kette $\tilde{P}$ ist ergodisch genau dann, wenn die ursprüngliche Kette P ergodisch ist. In diesem Fall gilt: Wenn $\{q_i\}_{i \in S}$ die stationäre Verteilung der ursprünglichen Kette ist, dann gilt für die stationäre Verteilung $\{q_{[i_1 \dots i_k]}\}_{[i_1 \dots i_k] \in \tilde{S}}$ der expandierten Kette:*
>
> $$q_{[i_1 \dots i_k]} = q_{i_1} \cdot p_{i_1 i_2} p_{i_2 i_3} \cdots p_{i_{k-1} i_k} \tag{3.3}$$

Beweis:
Wenn die ursprüngliche Kette ergodisch ist, dann gibt es ein $n > 0$, so daß alle Einträge $p_{ij}^{(n)}$ von P^n positiv sind. Dann ist

$$p_{[i_1\ldots i_k][j_1\ldots j_k]}^{(n+k-1)} = p_{i_k j_1}^{(n)} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k}$$

ebenfalls positiv, da $p_{j_1 j_2} \cdots p_{j_{k-1} j_k} > 0$ für alle $[j_1 \ldots j_k] \in \tilde{S}$ nach Definition der Zustände der expandierten Kette. Ebenso folgt, daß $\tilde{P}$ nicht ergodisch sein kann, wenn P es nicht ist.

Sei P ergodisch. Die Ausdrücke $q_{i_1} \cdot p_{i_1 i_2} p_{i_2 i_3} \cdots p_{i_{k-1} i_k}$ sind positiv, da q_{i_1} positiv ist und $p_{j_1 j_2} \cdots p_{j_{k-1} j_k}$ nach Definition auch positiv ist. Die Summe über alle Wahrscheinlichkeiten ist 1:

$$\begin{aligned}
\sum_{[i_1\ldots i_k]\in\tilde{S}} q_{[i_1\ldots i_k]} &= \sum_{i_1\ldots i_k} q_{i_1} \cdot p_{i_1 i_2} p_{i_2 i_3} \cdots p_{i_{k-1} i_k} \\
&= \sum_{i_2\ldots i_k} q_{i_2} \cdot p_{i_2 i_3} p_{i_3 i_4} \cdots p_{i_{k-1} i_k} \\
&\cdots \\
&= \sum_{i_k} q_{i_k} \\
&= 1
\end{aligned}$$

Es bleibt zu zeigen, daß die Verteilung stationär ist, d.h.

$$\sum_{[i_1\ldots i_k]} q_{[i_1\ldots i_k]} \cdot p_{[i_1\ldots i_k][j_1\ldots j_k]} = q_{[j_1\ldots j_k]} \qquad \text{für alle } [j_1 \ldots j_k] \in \tilde{S}$$

Dies gilt wegen

$$\begin{aligned}
\sum_{[i_1\ldots i_k]} q_{[i_1\ldots i_k]} \cdot p_{[i_1\ldots i_k][j_1\ldots j_k]} &= \sum_{i_1\ldots i_k} \left(q_{i_1} \cdot p_{i_1 i_2} p_{i_2 i_3} \cdots p_{i_{k-1} i_k} \cdot p_{i_k j_k} \cdot \delta_{[i_2\ldots i_k][j_1\ldots j_{k-1}]}\right) \\
&= \sum_{i_2\ldots i_k} q_{i_2} \cdot p_{i_2 i_3} \cdots p_{i_{k-1} i_k} \cdot p_{i_k j_k} \cdot \delta_{[i_2\ldots i_k][j_1\ldots j_{k-1}]} \\
&= q_{j_1} \cdot p_{j_1 j_2} p_{j_2 j_3} \cdots p_{j_{k-1} j_k} \\
&= q_{[j_1\ldots j_k]}
\end{aligned}$$

∎

Als nächstes benötigen wir die Matrix $\tilde{M}$ der mittleren Erst-Zugangs-Zeiten. Sie wird mit Hilfe der Fundamentalmatrix einer ergodischen Kette berechnet. Zunächst drücken wir die Fundamentalmatrix der expandierten Kette durch die Fundamentalmatrix der ursprünglichen Kette aus:

Lemma 3.5 :

Die Fundamentalmatrix $\tilde{Z}$ der expandierten Markov-Kette ist gegeben durch

$$\begin{aligned} z_{[i_1 \dots i_k][j_1 \dots j_k]} &= \delta_{[i_1 \dots i_k][j_1 \dots j_k]} + p_{j_1 j_2} \cdots p_{j_{k-1} j_k} (z_{i_k j_1} - (k-1) q_{j_1}) \\ &\quad + \sum_{n=1}^{k-2} (p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]}) \end{aligned} \tag{3.4}$$

Beweis:
Für die Übergangswahrscheinlichkeiten in mehreren Schritten gilt

$$p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]} = p^{(n-k+1)}_{i_k j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k} \text{ für } n \geq k,$$

da $k-1$ Schritte „innerhalb“ des Zustands $[j_1 \dots j_k]$ gemacht werden (im Sinne der ursprünglichen Kette). Also stehen nur $n-(k-1)$ Schritte für den Weg von i_k nach j_1 zur Verfügung. Für $n < k$ müssen sich die Tupel teilweise überlappen, nämlich um genau $k-n$ Komponenten:

$$\begin{aligned} p^{(k-1)}_{[i_1 \dots i_k][j_1 \dots j_k]} &= \delta_{i_k j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k} \\ p^{(k-2)}_{[i_1 \dots i_k][j_1 \dots j_k]} &= \delta_{i_{k-1} j_1} \delta_{i_k j_2} \cdot p_{j_2 j_3} \cdots p_{j_{k-1} j_k} \\ &\cdots \\ p^{(2)}_{[i_1 \dots i_k][j_1 \dots j_k]} &= \delta_{i_3 j_1} \cdots \delta_{i_k j_{k-2}} \cdot p_{j_{k-2} j_{k-1}} p_{j_{k-1} j_k} \\ p^{(1)}_{[i_1 \dots i_k][j_1 \dots j_k]} &= \delta_{i_2 j_1} \cdots \delta_{i_k j_{k-1}} \cdot p_{j_{k-1} j_k} \\ p^{(0)}_{[i_1 \dots i_k][j_1 \dots j_k]} &= \delta_{i_1 j_1} \cdots \delta_{i_k j_k} \end{aligned}$$

Der Fall $n = k-1$ wird auch durch den ersten Ausdruck dargestellt, wenn man die Konvention $p^{(0)}_{ij} = \delta_{ij}$ benutzt.
Die Komponenten der Fundamentalmatrix lassen sich gemäß Anhang A (Formel (A.1)) folgendermaßen schreiben:

$$\begin{aligned} z_{[i_1 \dots i_k][j_1 \dots j_k]} &= \delta_{[i_1 \dots i_k][j_1 \dots j_k]} + \sum_{n=1}^{\infty} (p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]} - q_{[j_1 \dots j_k]}) \\ &= \delta_{[i_1 \dots i_k][j_1 \dots j_k]} + \sum_{n=k-1}^{\infty} (p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]} - q_{[j_1 \dots j_k]}) \\ &\quad + \sum_{n=1}^{k-2} (p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]} - q_{[j_1 \dots j_k]}) \\ &= \delta_{[i_1 \dots i_k][j_1 \dots j_k]} + \sum_{n=k-1}^{\infty} (p^{(n-k+1)}_{i_k j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k} - q_{j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k}) \end{aligned}$$

$$
\begin{aligned}
&\quad - (k-2) \cdot q_{[j_1 \dots j_k]} + \sum_{n=1}^{k-2} p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]} \\
&= \delta_{[i_1 \dots i_k][j_1 \dots j_k]} + p_{j_1 j_2} \cdots p_{j_{k-1} j_k} \sum_{n=0}^{\infty} (p^{(n)}_{i_k j_1} - q_{j_1}) \\
&\quad - (k-2) \cdot q_{j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k} + \sum_{n=1}^{k-2} p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]} \\
&= \delta_{[i_1 \dots i_k][j_1 \dots j_k]} + p_{j_1 j_2} \cdots p_{j_{k-1} j_k} (z_{i_k j_1} - (k-1) q_{j_1}) \\
&\quad + \sum_{n=1}^{k-2} p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]}
\end{aligned}
$$

■

Speziell gilt:

$$
\begin{array}{lcll}
k = 2 & : & z_{[i_1 i_2][j_1 j_2]} & = \delta_{[i_1 i_2][j_1 j_2]} + p_{j_1 j_2} (z_{i_2 j_1} - q_{j_1}) \\
k = 3 & : & z_{[i_1 i_2 i_3][j_1 j_2 j_3]} & = \delta_{[i_1 i_2 i_3][j_1 j_2 j_3]} + p_{j_1 j_2} p_{j_2 j_3} (z_{i_3 j_1} - 2 q_{j_1}) + \delta_{i_2 j_1} \delta_{i_3 j_2} p_{j_2 j_3} \\
k = 4 & : & z_{[i_1 i_2 i_3 i_4][j_1 j_2 j_3 j_4]} & = \delta_{[i_1 i_2 i_3 i_4][j_1 j_2 j_3 j_4]} + p_{j_1 j_2} p_{j_2 j_3} p_{j_3 j_4} (z_{i_4 j_1} - 3 q_{j_1}) \\
 & & & \quad + \delta_{i_2 j_1} \delta_{i_3 j_2} \delta_{i_4 j_3} p_{j_3 j_4} + \delta_{i_3 j_1} \delta_{i_4 j_2} p_{j_2 j_3} p_{j_3 j_4}
\end{array}
$$

Satz 3.6:

Die Matrix $\tilde{M}$ der mittleren Erst-Zugangs-Zeiten einer expandierten Markov-Kette ist gegeben durch

$$
\begin{aligned}
m_{[i_1 \dots i_k][j_1 \dots j_k]} &= \frac{1 - \sum_{n=1}^{k-2} (p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]} - p^{(n)}_{[j_1 \dots j_k][j_1 \dots j_k]})}{q_{j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k}} \\
&\quad - \frac{z_{i_k j_1} - z_{j_k j_1}}{q_{j_1}}
\end{aligned}
\tag{3.5}
$$

Beweis:
Die Komponenten von $\tilde{M}$ lassen sich gemäß Anhang A (Formel (A.4)) schreiben als

$$
\begin{aligned}
m_{[i_1 \dots i_k][j_1 \dots j_k]} &= (\delta_{[i_1 \dots i_k][j_1 \dots j_k]} - z_{[i_1 \dots i_k][j_1 \dots j_k]} + z_{[j_1 \dots j_k][j_1 \dots j_k]}) / q_{[j_1 \dots j_k]} \\
&= \Big(\delta_{[i_1 \dots i_k][j_1 \dots j_k]} - \delta_{[i_1 \dots i_k][j_1 \dots j_k]} - p_{j_1 j_2} \cdots p_{j_{k-1} j_k} (z_{i_k j_1} - (k-1) q_{j_1}) \\
&\quad - \sum_{n=1}^{k-2} p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]} + 1 + p_{j_1 j_2} \cdots p_{j_{k-1} j_k} (z_{j_k j_1} - (k-1) q_{j_1}) \\
&\quad + \sum_{n=1}^{k-2} p^{(n)}_{[j_1 \dots j_k][j_1 \dots j_k]} \Big) / (q_{j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k})
\end{aligned}
$$

$$= \frac{1 - p_{j_1 j_2} \cdots p_{j_{k-1} j_k}(z_{i_k j_1} - z_{j_k j_1}) - \sum_{n=1}^{k-2}(p^{(n)}_{[i_1 \ldots i_k][j_1 \ldots j_k]} - p^{(n)}_{[j_1 \ldots j_k][j_1 \ldots j_k]})}{q_{j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k}}$$

$$= \frac{1 - \sum_{n=1}^{k-2}(p^{(n)}_{[i_1 \ldots i_k][j_1 \ldots j_k]} - p^{(n)}_{[j_1 \ldots j_k][j_1 \ldots j_k]})}{q_{j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k}} - \frac{z_{i_k j_1} - z_{j_k j_1}}{q_{j_1}}$$

■

Speziell:

$$k = 2 \quad : \quad m_{[i_1 i_2][j_1 j_2]} = \frac{1}{q_{j_1} p_{j_1 j_2}} - \frac{z_{i_2 j_1} - z_{j_2 j_1}}{q_{j_1}}$$

$$k = 3 \quad : \quad m_{[i_1 i_2 i_3][j_1 j_2 j_3]} = \frac{1 - (\delta_{i_2 j_1}\delta_{i_3 j_2} p_{j_2 j_3} - \delta_{j_2 j_1}\delta_{j_3 j_2} p_{j_2 j_3})}{q_{j_1} p_{j_1 j_2} p_{j_2 j_3}} - \frac{z_{i_3 j_1} - z_{j_3 j_1}}{q_{j_1}}$$

$$\begin{aligned} k = 4 \quad : \quad m_{[i_1 i_2 i_3 i_4][j_1 j_2 j_3 j_4]} &= \frac{1 - (\delta_{i_2 j_1}\delta_{i_3 j_2}\delta_{i_4 j_3} p_{j_3 j_4} - \delta_{j_2 j_1}\delta_{j_3 j_2}\delta_{j_4 j_3} p_{j_3 j_4})}{q_{j_1} p_{j_1 j_2} p_{j_2 j_3} p_{j_3 j_4}} \\ &\quad - \frac{\delta_{i_3 j_1}\delta_{i_4 j_2} p_{j_2 j_3} p_{j_3 j_4} - \delta_{j_3 j_1}\delta_{j_4 j_2} p_{j_2 j_3} p_{j_3 j_4}}{q_{j_1} p_{j_1 j_2} p_{j_2 j_3} p_{j_3 j_4}} \\ &\quad - \frac{z_{i_4 j_1} - z_{j_4 j_1}}{q_{j_1}} \end{aligned}$$

Was passiert mit der expandierten Kette, wenn man die Zeit umdreht? Es gilt, daß die rückwärtslaufende Folge der expandierten Kette gerade die expandierte Kette der rückwärtslaufenden ursprünglichen Kette ist. Wie vorher werden die entsprechenden Wahrscheinlichkeiten mit einem Dach versehen.

Satz 3.7:

Für die Übergangswahrscheinlichkeiten der rückwärtslaufenden expandierten Kette gilt:

$$\hat{p}_{[i_1 \ldots i_k][j_1 \ldots j_k]} = \hat{p}_{i_1 j_1} \cdot \delta_{[i_1 \ldots i_{k-1}][j_2 \ldots j_k]} \tag{3.6}$$

Beweis:
Für die Übergangswahrscheinlichkeiten einer ergodischen Markov-Kette bei Rückwärtsschritten gilt

$$\begin{aligned} \hat{p}_{[i_1 \ldots i_k][j_1 \ldots j_k]} &= \frac{q_{[j_1 \ldots j_k]} \cdot p_{[j_1 \ldots j_k][i_1 \ldots i_k]}}{q_{[i_1 \ldots i_k]}} \\ &= \frac{q_{j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k} \cdot p_{j_k i_k} \cdot \delta_{[j_2 \ldots j_k][i_1 \ldots i_{k-1}]}}{q_{i_1} \cdot p_{i_1 i_2} \cdots p_{i_{k-1} i_k}} \\ &= \frac{q_{j_1} \cdot p_{j_1 i_1}}{q_{i_1}} \cdot \delta_{[i_1 \ldots i_{k-1}][j_2 \ldots j_k]} \end{aligned}$$

$$= \hat{p}_{i_1 j_1} \cdot \delta_{[i_1 \ldots i_{k-1}][j_2 \ldots j_k]}$$

Das Fenster der Länge k wird also um eine Position nach *links* auf der ursprünglichen Kette geschoben, und das Element i_k tritt rechts aus dem Fenster aus. ■

Für die Übergangswahrscheinlichkeiten in n Schritten finden wir für $n \geq k$

$$\begin{aligned}
\hat{p}^{(n)}_{[i_1 \ldots i_k][j_1 \ldots j_k]} &= \hat{p}^{(n-k+1)}_{i_1 j_k} \cdot \hat{p}_{j_k j_{k-1}} \cdot \ldots \cdot \hat{p}_{j_2 j_1} \\
&= \frac{q_{j_k}}{q_{i_1}} p^{(n-k+1)}_{j_k i_1} \cdot \frac{q_{j_{k-1}}}{q_{j_k}} p_{j_{k-1} j_k} \cdots \frac{q_{j_1}}{q_{j_2}} p_{j_1 j_2} \\
&= \frac{q_{j_1}}{q_{i_1}} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k} \cdot p^{(n-k+1)}_{j_k i_1}
\end{aligned}$$

und für $n < k$ wieder teilweise Überlappung der Index-Tupel

$$\begin{aligned}
\hat{p}^{(k-1)}_{[i_1 \ldots i_k][j_1 \ldots j_k]} &= \frac{q_{j_1}}{q_{i_1}} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k} \cdot \delta_{j_k i_1} \\
\hat{p}^{(k-2)}_{[i_1 \ldots i_k][j_1 \ldots j_k]} &= \frac{q_{j_1}}{q_{i_1}} \cdot p_{j_1 j_2} \cdots p_{j_{k-2} j_{k-1}} \cdot \delta_{j_{k-1} i_1} \delta_{j_k i_2} \\
&\ldots \\
\hat{p}^{(2)}_{[i_1 \ldots i_k][j_1 \ldots j_k]} &= \frac{q_{j_1}}{q_{i_1}} \cdot p_{j_1 j_2} \cdot p_{j_2 j_3} \cdot \delta_{j_3 i_1} \cdots \delta_{j_k i_{k-2}} \\
\hat{p}^{(1)}_{[i_1 \ldots i_k][j_1 \ldots j_k]} &= \frac{q_{j_1}}{q_{i_1}} \cdot p_{j_1 j_2} \cdot \delta_{j_2 i_1} \cdots \delta_{j_k i_{k-1}} \\
\hat{p}^{(0)}_{[i_1 \ldots i_k][j_1 \ldots j_k]} &= \delta_{j_1 i_1} \cdots \delta_{j_k i_k}
\end{aligned} \tag{3.7}$$

Als nächstes berechnen wir die Fundamentalmatrix $\hat{Z}$ und die Matrix der mittleren Erst-Eintritts-Zeiten $\hat{M}$ für die rückwärtslaufende expandierte Kette. Wir führen diese Werte zurück auf Ausdrücke der ursprünglichen (vorwärtslaufenden, nicht expandierten) Kette. Damit die Notation nicht zu kompliziert wird, lassen wir die Wahrscheinlichkeiten $\hat{p}^{(n)}_{[i_1 \ldots i_k][j_1 \ldots j_k]}$ für $n < k-1$ stehen. Sie sind gemäß (3.7) gegeben und können für konkrete Anwendungen leicht eingesetzt werden. In unserem Fall, nämlich die Berechnung der erwarteten Suchzeit der Simple-k-Nach-Vorne-Schieben-Regel bei abhängigen Zugriffen, werden wir dies tun.

Lemma 3.8:

Die Fundamentalmatrix $\hat{Z}$ der rückwärtslaufenden expandierten Kette ist gegeben durch

$$\begin{aligned}
\hat{z}_{[i_1 \ldots i_k][j_1 \ldots j_k]} &= \delta_{[i_1 \ldots i_k][j_1 \ldots j_k]} + p_{j_1 j_2} \cdots p_{j_{k-1} j_k} \cdot \left(\frac{q_{j_1}}{q_{i_1}} \cdot z_{j_k i_1} - (k-1) q_{j_1} \right) \\
&\quad + \sum_{n=1}^{k-2} \hat{p}^{(n)}_{[i_1 \ldots i_k][j_1 \ldots j_k]}
\end{aligned} \tag{3.8}$$

Beweis:

$$
\begin{aligned}
\hat{z}_{[i_1\ldots i_k][j_1\ldots j_k]} &= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + \sum_{n=1}^{\infty}\left(\hat{p}^{(n)}_{[i_1\ldots i_k][j_1\ldots j_k]} - q_{[j_1\ldots j_k]}\right) \\
&= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + \sum_{n=k-1}^{\infty}\left(\frac{q_{j_1}}{q_{i_1}} \cdot p_{j_1 j_2}\cdots p_{j_{k-1}j_k} p^{(n-k+1)}_{j_k i_1} - q_{j_1} p_{j_1 j_2}\cdots p_{j_{k-1}j_k}\right) \\
&\quad + \sum_{n=1}^{k-2}(\hat{p}^{(n)}_{[i_1\ldots i_k][j_1\ldots j_k]} - q_{[j_1\ldots j_k]}) \\
&= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + \frac{q_{j_1}}{q_{i_1}} \cdot p_{j_1 j_2}\cdots p_{j_{k-1}j_k} \sum_{n=0}^{\infty}(p^{(n)}_{j_k i_1} - q_{i_1}) \\
&\quad - (k-2) q_{j_1} p_{j_1 j_2}\cdots p_{j_{k-1}j_k} + \sum_{n=1}^{k-2}\hat{p}^{(n)}_{[i_1\ldots i_k][j_1\ldots j_k]} \\
&= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + \frac{q_{j_1}}{q_{i_1}} \cdot p_{j_1 j_2}\cdots p_{j_{k-1}j_k}(z_{j_k i_1} - q_{i_1}) \\
&\quad - (k-2) q_{j_1} \cdot p_{j_1 j_2}\cdots p_{j_{k-1}j_k} + \sum_{n=1}^{k-2}\hat{p}^{(n)}_{[i_1\ldots i_k][j_1\ldots j_k]} \\
&= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + p_{j_1 j_2}\cdots p_{j_{k-1}j_k}\left(\frac{q_{j_1}}{q_{i_1}} \cdot z_{j_k i_1} - (k-1) q_{j_1}\right) \\
&\quad + \sum_{n=1}^{k-2}\hat{p}^{(n)}_{[i_1\ldots i_k][j_1\ldots j_k]}
\end{aligned}
$$

■

Lemma 3.9:

Die Matrix $\hat{M}$ der rückwärtslaufenden expandierten Kette ist gegeben durch

$$
\hat{m}_{[i_1\ldots i_k][j_1\ldots j_k]} = \frac{1 - \sum_{n=1}^{k-2}(\hat{p}^{(n)}_{[i_1\ldots i_k][j_1\ldots j_k]} - \hat{p}^{(n)}_{[j_1\ldots j_k][j_1\ldots j_k]})}{q_{j_1} \cdot p_{j_1 j_2}\cdots p_{j_{k-1}j_k}} - \frac{z_{j_k i_1}}{q_{i_1}} + \frac{z_{j_k j_1}}{q_{j_1}}
$$

Beweis:

$$
\begin{aligned}
\hat{m}_{[i_1\ldots i_k][j_1\ldots j_k]} &= (\delta_{[i_1\ldots i_k][j_1\ldots j_k]} - \hat{z}_{[i_1\ldots i_k][j_1\ldots j_k]} + \hat{z}_{[j_1\ldots j_k][j_1\ldots j_k]})/q_{[j_1\ldots j_k]} \\
&= \Big(\delta_{[i_1\ldots i_k][j_1\ldots j_k]} - \delta_{[i_1\ldots i_k][j_1\ldots j_k]} - p_{j_1 j_2}\cdots p_{j_{k-1}j_k}(\frac{q_{j_1}}{q_{i_1}} z_{j_k i_1} - (k-1) q_{j_1}) \\
&\quad - \sum_{n=1}^{k-2}\hat{p}^{(n)}_{[i_1\ldots i_k][j_1\ldots j_k]} + 1 + p_{j_1 j_2}\cdots p_{j_{k-1}j_k}(z_{j_k j_1} - (k-1) q_{j_1}) \\
&\quad + \sum_{n=1}^{k-2}\hat{p}^{(n)}_{[j_1\ldots j_k][j_1\ldots j_k]}\Big)/(q_{j_1} \cdot p_{j_1 j_2}\cdots p_{j_{k-1}j_k})
\end{aligned}
$$

$$= \frac{1 + p_{j_1 j_2} \cdots p_{j_{k-1} j_k}(z_{j_k j_1} - \frac{q_{j_1}}{q_{i_1}} \cdot z_{j_k i_1}) - \sum_{n=1}^{k-2}(\hat{p}^{(n)}_{[i_1 \ldots i_k][j_1 \ldots j_k]} - \hat{p}^{(n)}_{[j_1 \ldots j_k][j_1 \ldots j_k]})}{q_{j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k}}$$

$$= \frac{1 - \sum_{n=1}^{k-2}(\hat{p}^{(n)}_{[i_1 \ldots i_k][j_1 \ldots j_k]} - \hat{p}^{(n)}_{[j_1 \ldots j_k][j_1 \ldots j_k]})}{q_{j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k}} - \frac{z_{j_k i_1}}{q_{i_1}} + \frac{z_{j_k j_1}}{q_{j_1}}$$

■

3.4 Abhängige Zugriffe

In diesem Abschnitt werden die k-Mal-Hintereinander-Strategien für die Nach-Vorne-Schieben-Regel analysiert.

3.4.1 Simple-k-Nach-Vorne-Schieben-Regel

Für die Untersuchung der Simple-k-Move-to-Front–Regel bei abhängigen Zugriffen machen wir folgende Beobachtung:

Lemma 3.10 (Strukturlemma für die Simple-k-Nach-Vorne-Schieben-Regel)
Das Element j steht vor dem Element i, wenn die letzte Folge von k aufeinanderfolgenden Zugriffen auf j nach der letzten Folge von k aufeinanderfolgenden Zugriffen auf i aufgetreten ist.

Umformulierung:

Beim Zugriff auf l steht j vor i, wenn wir bei l startend die Anfrage-Folge zurückgehen und auf k aufeinanderfolgende Zugriffe auf j treffen, bevor k aufeinanderfolgende Zugriffe auf i aufgetreten sind.

Wenn also gerade auf i zugegriffen wird, dann suchen wir nach der asymptotischen Wahrscheinlichkeit für Situationen der Art

$\ldots\ldots iiii \ldots\ldots jjjj \ldots\ldots l_1 l_2 l_3 i$

wenn man die Folge von links nach rechts darstellt ($k = 4$). Dabei durchlaufen l_1, l_2, l_3 alle möglichen Werte.

3.4.1.1 Herleitung der erwarteten Suchzeit

Wir berechnen zuerst die asymptotische Wahrscheinlichkeit, daß ein Tupel $[jj \ldots j]$ vor einem Tupel $[ii \ldots i]$ auftritt, wenn man die expandierte Kette von $[l_1 \ldots l_{k-1} i]$

aus rückwärtsverfolgt. Diese Wahrscheinlichkeit sei mit $b([j\dots j],[i\dots i]\mid[l_1\dots l_{k-1}i])$ bezeichnet.

Lemma 3.11:

Es gilt

$$b([j\dots j],[i\dots i]\mid[l_1\dots l_{k-1}i]) = \frac{A}{B}$$

mit dem Zähler

$$\begin{aligned} A \;=\;& q_j p_{jj}^{k-1}\sum_{n=0}^{k-2} p_{ii}^n \;-\; q_j p_{jj}^{k-1}\sum_{n=1}^{k-2} p_{ii}^n\cdot\delta_{l_1 i}\cdots\delta_{l_{k-n}i} \;+\; q_i p_{ii}^{k-1}\sum_{n=1}^{k-2} p_{jj}^n\cdot\delta_{l_1 j}\cdots\delta_{l_{k-n}j}\\ &+ q_i q_j p_{ii}^{k-1} p_{jj}^{k-1}\left(m_{il_1} - m_{jl_1} - \frac{\delta_{il_1}-\delta_{jl_1}}{q_{l_1}} + m_{ji}\right)\end{aligned}$$

und dem Nenner

$$B \;=\; q_i p_{ii}^{k-1}\sum_{n=0}^{k-2} p_{jj}^n \;+\; q_j p_{jj}^{k-1}\sum_{n=0}^{k-2} p_{ii}^n \;+\; q_i q_j p_{ii}^{k-1} p_{jj}^{k-1}\cdot(m_{ij}+m_{ji})$$

Beweis:
Die Wahrscheinlichkeit, von $[l_1\dots l_{k-1}i]$ startend auf $[j\dots j]$ zu treffen, bevor $[i\dots i]$ auftritt, kann nach Lemma 1.13 aus Kapitel 1 berechnet werden. Die darin auftretenden Werte werden gemäß Lemma 3.9 eingesetzt.

$$\begin{aligned} & b([j\dots j],[i\dots i]\mid[l_1\dots l_{k-1}i])\\ &= \frac{\hat m_{[l_1\dots l_{k-1}i][i\dots i]} + \hat m_{[i\dots i][j\dots j]} - \hat m_{[l_1\dots l_{k-1}i][j\dots j]}}{\hat m_{[i\dots i][j\dots j]} + \hat m_{[j\dots j][i\dots i]}}\\ &= \frac{A'}{B'}\end{aligned}$$

mit

$$\begin{aligned} A' \;=\;& \frac{1-\sum_{n=1}^{k-2}(\hat p^{(n)}_{[l_1\dots l_{k-1}i][i\dots i]} - \hat p^{(n)}_{[i\dots i][i\dots i]})}{q_i p_{ii}^{k-1}} - \frac{z_{il_1}}{q_{l_1}} + \frac{z_{ii}}{q_i}\\ &+ \frac{1-\sum_{n=1}^{k-2}(\hat p^{(n)}_{[i\dots i][j\dots j]} - \hat p^{(n)}_{[j\dots j][j\dots j]})}{q_j p_{jj}^{k-1}} - \frac{z_{ji}}{q_i} + \frac{z_{jj}}{q_j}\\ &- \frac{1-\sum_{n=1}^{k-2}(\hat p^{(n)}_{[l_1\dots l_{k-1}i][j\dots j]} - \hat p^{(n)}_{[j\dots j][j\dots j]})}{q_j p_{jj}^{k-1}} + \frac{z_{jl_1}}{q_{l_1}} - \frac{z_{jj}}{q_j}\end{aligned}$$

und

$$\begin{aligned}
B' &= \frac{1-\sum_{n=1}^{k-2}(\hat{p}^{(n)}_{[i\ldots i][j\ldots j]}-\hat{p}^{(n)}_{[j\ldots j][j\ldots j]})}{q_j p_{jj}^{k-1}} - \frac{z_{ji}}{q_i} + \frac{z_{jj}}{q_j} \\
&\quad + \frac{1-\sum_{n=1}^{k-2}(\hat{p}^{(n)}_{[j\ldots j][i\ldots i]}-\hat{p}^{(n)}_{[i\ldots i][i\ldots i]})}{q_i p_{ii}^{k-1}} - \frac{z_{ij}}{q_j} + \frac{z_{ii}}{q_i}
\end{aligned}$$

Durch Erweitern des Bruches mit $q_i q_j p_{ii}^{k-1} p_{jj}^{k-1}$ und Beachten von $i \neq j$ sowie $\hat{p}^{(n)}_{[i\ldots i][i\ldots i]} = p_{ii}^n$ für $n < k$ folgt

$$\begin{aligned}
A &= q_i q_j p_{ii}^{k-1} p_{jj}^{k-1} \cdot A' \\
&= q_j p_{jj}^{k-1}(\sum_{n=0}^{k-2} p_{ii}^n) - q_j p_{jj}^{k-1} \sum_{n=1}^{k-2} \hat{p}^{(n)}_{[l_1\ldots l_{k-1}i][i\ldots i]} + q_i p_{ii}^{k-1} \sum_{n=1}^{k-2} \hat{p}^{(n)}_{[l_1\ldots l_{k-1}i][i\ldots i]} \\
&\quad - q_i q_j p_{ii}^{k-1} p_{jj}^{k-1}(\frac{z_{il_1}+z_{jl_1}}{q_{l_1}} + \frac{z_{ji}-z_{ii}}{q_i}) \\
B &= q_i q_j p_{ii}^{k-1} p_{jj}^{k-1} \cdot B' \\
&= q_i p_{ii}^{k-1} \sum_{n=0}^{k-2} p_{jj}^n + q_j p_{jj}^{k-1} \sum_{n=0}^{k-2} p_{ii}^n - q_i q_j p_{ii}^{k-1} p_{jj}^{k-1}(\frac{z_{ji}-z_{ii}}{q_i} + \frac{z_{ij}-z_{jj}}{q_j})
\end{aligned}$$

Das Ergebnis folgt durch Ausnutzen von $m_{ij} = (\delta_{ij} - z_{ij} + z_{jj})/q_j$ und $i \neq j$ sowie

$$\begin{aligned}
\hat{p}^{(n)}_{[l_1\ldots l_{k-1}i][i\ldots i]} &= \frac{q_i}{q_{l_1}} \cdot p_{ii}^n \cdot \delta_{l_1 i} \cdots \delta_{l_{k-n} i} \\
&= p_{ii}^n \cdot \delta_{l_1 i} \cdots \delta_{l_{k-n} i}
\end{aligned}$$

für $1 \leq n \leq k-1$, was sich aus den Formeln (3.7) ergibt. ∎

Die asymptotische Wahrscheinlichkeit, daß j vor i steht beim Zugriff auf i, erhält man, indem man über alle möglichen Vorgänger $l_1, \ldots, l_{k-1}$ von i summiert, gewichtet mit der Wahrscheinlichkeit ihres Auftretens.

Lemma 3.12:

Für die Simple-k-Nach-Vorne-Schieben-Regel gilt bei abhängigen Zugriffen

$$b_k(j,i|i) \quad = \quad \frac{A}{B} \tag{3.9}$$

mit dem Zähler

$$\begin{aligned} A \quad = \quad & q_j p_{jj}^{k-1}\Big(\sum_{n=0}^{k-2} p_{ii}^n \; + \; p_{ii}^{k-1}\sum_{n=1}^{k-1}(p_{ji}^{(n)} - p_{ii}^{(n)}) \\ & \qquad + p_{ii}^{k-1}\Big(q_i m_{ji} + \sum_{l_1} q_{l_1} p_{l_1 i}^{(k-1)}(m_{il_1} - m_{jl_1})\Big)\Big) \end{aligned}$$

und dem Nenner wie in Lemma 3.11

$$B \quad = \quad q_i p_{ii}^{k-1}\sum_{n=0}^{k-2} p_{jj}^n \; + \; q_j p_{jj}^{k-1}\sum_{n=0}^{k-2} p_{ii}^n \; + \; q_i q_j p_{ii}^{k-1} p_{jj}^{k-1}(m_{ij} + m_{ji})$$

Dabei ist die n-te Potenz p_{ij}^n zu unterscheiden von der Übergangswahrscheinlichkeit in n Schritten $p_{ij}^{(n)}$.

Beweis:
Die Wahrscheinlichkeit des Auftretens von $[l_1 \dots l_{k-1} i]$ ist die Wahrscheinlichkeit, von i aus rückwärts zu gehen und nacheinander auf $l_{k-1}, \dots, l_1$ zu stoßen:

$$\begin{aligned} \hat{p}_{il_{k-1}} \cdot \hat{p}_{l_{k-1}l_{k-2}} \cdots \hat{p}_{l_2 l_1} \quad &= \quad \frac{q_{l_{k-1}}}{q_i} p_{l_{k-1}i} \cdot \frac{q_{l_{k-2}}}{q_{l_{k-1}}} p_{l_{k-2}l_{k-1}} \cdots \frac{q_{l_1}}{q_{l_2}} p_{l_1 l_2} \\ &= \quad \frac{q_{l_1}}{q_i} \cdot p_{l_1 l_2} \cdots p_{l_{k-2}l_{k-1}} \cdot p_{l_{k-1}i} \end{aligned}$$

Dies setzen wir ein:

$$b_k(j,i|i) \quad = \quad \frac{1}{q_i} \sum_{l_1 \dots l_{k-1}} q_{l_1} p_{l_1 l_2} \cdots p_{l_{k-2}l_{k-1}} p_{l_{k-1}i} \cdot b([j \dots j],[i \dots i] \mid [l_1 \dots l_{k-1} i])$$

Wir betrachten die Summationen über die einzelnen Summanden des Zählers der Formel in Lemma 3.11 getrennt und finden

$$\begin{aligned} & \sum_{l_1 \dots l_{k-1}} q_{l_1} p_{l_1 l_2} \cdots p_{l_{k-1}i} \cdot \sum_{n=1}^{k-2} p_{jj}^n \delta_{l_1 j} \cdots \delta_{l_{k-n}j} \\ = \quad & q_j \sum_{l_2 \dots l_{k-1}} p_{jl_2} \cdots p_{l_{k-1}i} \cdot \sum_{n=1}^{k-2} p_{jj}^n \delta_{l_2 j} \cdots \delta_{l_{k-n}j} \\ = \quad & q_j \cdot \Big(\sum_{l_2 \dots l_{k-1}} p_{jl_2} \cdots p_{l_{k-1}i} \cdot p_{jj} \cdot \delta_{l_2 j} \cdots \delta_{l_{k-1}j} \end{aligned}$$

$$
\begin{aligned}
& + \sum_{l_2 \ldots l_{k-1}} p_{jl_2} \cdots p_{l_{k-1}i} \cdot p_{jj}^2 \cdot \delta_{l_2 j} \cdots \delta_{l_{k-2}j} \\
& \cdots \\
& + \sum_{l_2 \ldots l_{k-1}} p_{jl_2} \cdots p_{l_{k-1}i} \cdot p_{jj}^{k-2} \cdot \delta_{l_2 j}\Big) \\
= \quad & q_j \cdot \Big(p_{jj}^{k-1} \cdot p_{ji}^{(1)} \\
& + p_{jj}^{k-1} \cdot p_{ji}^{(2)} \\
& \cdots \\
& + p_{jj}^{k-1} \cdot p_{ji}^{(k-2)} \Big) \\
= \quad & q_j p_{jj}^{k-1} \sum_{n=1}^{k-2} p_{ji}^{(n)}
\end{aligned}
$$

sowie

$$
\begin{aligned}
& \sum_{l_1 \ldots l_{k-1}} q_{l_1} p_{l_1 l_2} \cdots p_{l_{k-1}i} \cdot (m_{ji} - \frac{\delta_{il_1} - \delta_{jl_1}}{q_{l_1}} + m_{il_1} - m_{jl_1}) \\
= \quad & \sum_{l_1} q_{l_1} p_{l_1 i}^{(k-1)} \cdot (m_{ji} + m_{il_1} - m_{jl_1}) - q_i p_{ii}^{(k-1)} \cdot (1/q_i) + q_j p_{ji}^{(k-1)} \cdot (1/q_j) \\
= \quad & q_i m_{ji} - p_{ii}^{(k-1)} + p_{ji}^{(k-1)} + \sum_{l_1} q_{l_1} p_{l_1 i}^{(k-1)} (m_{il_1} - m_{jl_1})
\end{aligned}
$$

Dabei haben wir die Eigenschaft der stationären Verteilung benutzt: $\sum_i q_i p_{ij} = q_j$. Durch Einsetzen und Zusammenfassen folgt das Ergebnis. ∎

Satz 3.13:

Die erwartete Suchzeit für die Simple-k-Nach-Vorne-Schieben-Regel bei einer Markov-Zugriffsfolge ist

$$\mu = 1 + \sum_{i \in S} \sum_{j \neq i} q_i \cdot b_k(j, i|i)$$

wobei $b_k(j, i|i)$ gemäß Lemma 3.12 gegeben ist.

Im Spezialfall unabhängiger Zugriffe erhalten wir das Ergebnis von Gonnet/Munro/Suwanda aus Lemma 3.3 wieder.

Korollar 3.14:

Bei unabhängigen Zugriffen gilt für die Formeln (3.1) und (3.9) und für beliebige k die Gleichheit für alle $i \neq j$.

$$b_k(j, i) = b_k(j, i|i)$$

Beweis:
Für unabhängige Zugriffe ist $p_{ij} = q_j$, und damit auch $p_{ij}^{(n)} = q_j$. Außerdem gilt $m_{ij} = 1/q_j$. Mit diesen Werten erhält man für $b_k(j, i|i)$:
Für den Nenner

$$\begin{aligned} B &= q_i^k \sum_{n=0}^{k-2} q_j^n \;+\; q_j^k \sum_{n=0}^{k-2} q_i^n \;+\; q_i^k q_j^k (1/q_i + 1/q_j) \\ &= q_i^k \sum_{n=0}^{k-1} q_j^n \;+\; q_j^k \sum_{n=0}^{k-1} q_i^n \end{aligned}$$

den Nenner von $b_k(j, i)$ und für den Zähler

$$\begin{aligned} A &= q_j^k \Big(\sum_{n=0}^{k-2} q_i^n \;+\; q_i^{k-1} \sum_{n=1}^{k-1} (q_i - q_i) \;+\; q_i^{k-1} \Big(1 + \sum_{l_1} q_{l_1} q_i (1/q_{l_1} - 1/q_{l_1}) \Big) \Big) \\ &= q_j^k \Big(\sum_{n=0}^{k-2} q_i^n \;+\; q_i^{k-1} \Big) \\ &= q_j^k \sum_{n=0}^{k-1} q_i^n \end{aligned}$$

den Zähler von $b_k(j, i)$. ■

3.4.1.2 Beispiel

Wir betrachten die Zugriffsfolge von Beispiel 1 (Seite 20). Sei $P = (p_{ij})$ die Übergangsmatrix mit den Einträgen

$$p_{ij} = \left\{ \begin{array}{lcl} \alpha & : & i = j \\ \beta & : & i \neq j \end{array} \right\} \tag{3.10}$$

Dann ist mit $\beta = (1 - \alpha)/(n - 1)$

$$\begin{aligned} p_{ij}^{(m)} &= \frac{1}{n} \left(1 + (\delta_{ij} n - 1) \left(\frac{n\alpha - 1}{n - 1} \right)^m \right) \\ m_{ij} &= \frac{n - 1 + \delta_{ij}(1 - n\alpha)}{1 - \alpha} \end{aligned}$$

Die stationäre Verteilung ist $q_i = 1/n$.
Für $b_k(j, i|i)$ berechnet man bei $i \neq j$:

$$\begin{aligned} A &= \frac{\alpha^{k-1}(1 - \alpha^k)}{n(1 - \alpha)} \\ B &= \frac{2\alpha^{k-1}(n - \alpha^{k-1})}{n^2(1 - \alpha)} \\ b_k(j, i \mid i) &= \frac{A}{B} = \frac{n(1 - \alpha^k)}{2(n - \alpha^{k-1})} \end{aligned}$$

Für die erwartete Suchzeit folgt

$$\begin{aligned}\mu &= 1 + \sum_i \sum_{j \neq i} q_i \cdot b_k(j, i \mid i) \\ &= 1 + \frac{n(n-1)(1-\alpha^k)}{2(n-\alpha^{k-1})}\end{aligned}$$

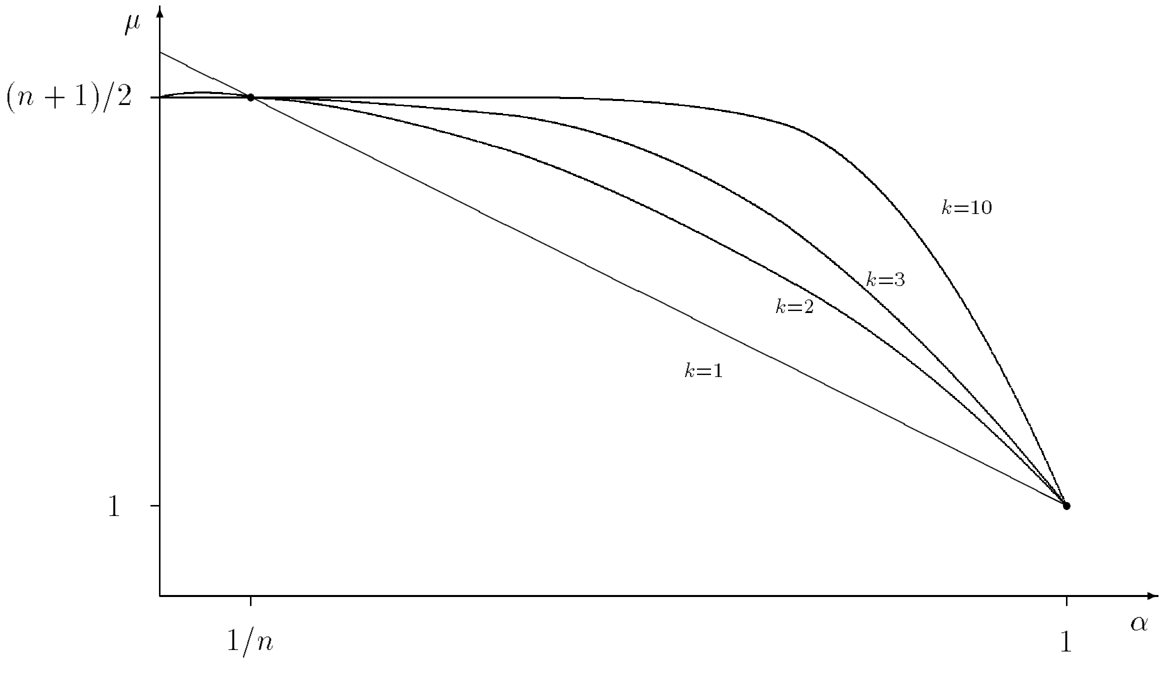
Satz 3.15 :

Für die erwartete Suchzeit der Simple-k-Nach-Vorne-Schieben-Regel bei der Zugriffsfolge (3.10) gilt

$$\mu = 1 + \frac{n(n-1)(1-\alpha^k)}{2(n-\alpha^{k-1})} \tag{3.11}$$

Interpretation:
Für $k = 1$ folgt wieder das bekannte Ergebnis $\mu = 1 + (1-\alpha)n/2$.
Bei $k > 1$ wird die Anpassung nach einem Wechsel der Zugriffe verzögert. Wenn das Übergangsverhalten träge ist ($\alpha > 1/n$), wirkt sich dies negativ aus. Wenn jeder Wechsel gleichwahrscheinlich ist ($\alpha = 1/n$), ist die erwartete Suchzeit $\mu = (n+1)/2$ für alle Werte k.
Im Grenzfall $k \to \infty$ wird die Anpassung unendlich lange verzögert. Für alle $\alpha < 1$ ist die erwartete Suchzeit $\mu = (n+1)/2$, da keine Umordnung gemacht wird.

Folgendes Diagramm gibt den qualitativen Verlauf an:



3.4.2 Batched-k-Nach-Vorne-Schieben-Regel

Bei der Batched-k-Regel werden die Zugriffe in Blöcke der Länge k eingeteilt. Die Aktion der Heuristik wird nur ausgeführt, wenn alle Zugriffe eines Blocks auf das gleiche Element gerichtet waren.
Wir formulieren wieder ein Strukturlemma, durch das die Regel charakterisiert wird.

Lemma 3.16 (Strukturlemma für die Batched-k-Nach-Vorne-Schieben-Regel)
Das Element j steht vor dem Element i, wenn der letzte Block von Zugriffen auf j nach dem letzten Block von Zugriffen auf i aufgetreten ist.

Umformulierung:
Beim Zugriff auf l steht j vor i, wenn wir beim aktuellen Block (der l enthält) startend die Anfrage-Folge zurückgehen und auf einen Block von Zugriffen auf j treffen, bevor ein Block von Zugriffen auf i aufgetreten ist.

Anstatt die Folge elementweise rückwärts zu laufen, können wir also blockweise rückwärts gehen. Anders als bei der Simple-k-Regel entstehen hier keine Probleme mit Überlappungen von Zustands-Tupeln.

3.4.2.1 Expansion der Kette

Für gegebenes $k > 1$ betrachten wir wieder die Kette $\tilde{P}$ mit der Zustandsmenge

$$\tilde{S} = \{[i_1 \dots i_k] \mid p_{i_1 i_2} \cdots p_{i_{k-1} i_k} > 0\}$$

Die Übergangswahrscheinlichkeiten sind jetzt

$$p_{[i_1 \dots i_k][j_1 \dots j_k]} = p_{i_k j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k}$$

und allgemein für Übergänge in n Schritten

$$p^{(n)}_{[i_1 \dots i_k][j_1 \dots j_k]} = p^{((n-1)k+1)}_{i_k j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k}$$

Man rechnet nach, daß für die stationären Wahrscheinlichkeiten wie vorher gilt

$$q_{[j_1 \dots j_k]} = q_{j_1} \cdot p_{j_1 j_2} \cdots p_{j_{k-1} j_k}$$

Wir definieren $\bar{P} = P^k$ als die Übergangsmatrix, die k Schritte zu einem Schritt zusammenfaßt. Die stationäre Verteilung der Kette $\bar{P}$ ist natürlich $(\bar{q}_j) = (q_j)$. Damit ist

$$\begin{aligned} p_{ij}^{((n-1)k+1)} &= \sum_l p_{il}^{((n-1)k)} p_{lj} \\ &= \sum_l \bar{p}_{il}^{(n-1)} p_{lj} \end{aligned}$$

Das macht es uns leichter, die Fundamentalmatrix $\tilde{Z}$ auszurechnen. Wir drücken sie aus mit Hilfe der Fundamentalmatrix $\bar{Z}$ der Kette $\bar{P}$.

$$\begin{aligned} z_{[i_1\ldots i_k][j_1\ldots j_k]} &= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + \sum_{n=1}^{\infty}(p^{(n)}_{[i_1\ldots i_k][j_1\ldots j_k]} - q_{[j_1\ldots j_k]}) \\ &= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + p_{j_1j_2}\cdots p_{j_{k-1}j_k} \cdot \sum_{n=1}^{\infty}\left(\sum_l \bar{p}_{i_kl}^{(n-1)} p_{lj_1} - q_{j_1}\right) \\ &= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + p_{j_1j_2}\cdots p_{j_{k-1}j_k} \cdot \sum_{n=1}^{\infty}\left(\sum_l \bar{p}_{i_kl}^{(n-1)} p_{lj_1} - \sum_l q_l p_{lj_1}\right) \\ &= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + p_{j_1j_2}\cdots p_{j_{k-1}j_k} \cdot \sum_{n=1}^{\infty}\left(\sum_l \left(\bar{p}_{i_kl}^{(n-1)} - q_l\right) p_{lj_1}\right) \\ &= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + p_{j_1j_2}\cdots p_{j_{k-1}j_k} \cdot \sum_l \left(\sum_{n=1}^{\infty} \left(\bar{p}_{i_kl}^{(n-1)} - q_l\right) p_{lj_1}\right) \\ &= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + p_{j_1j_2}\cdots p_{j_{k-1}j_k} \cdot \sum_l \left(\bar{z}_{i_kl} - q_l\right) p_{lj_1} \end{aligned}$$

Das Vertauschen der Summation ist erlaubt, weil die Summe der Grenzwerte konvergenter Folgen gerade der Grenzwert der Summe der Folgenglieder ist.

Für die erwarteten Erst-Eintritts-Zeiten m_{ij} folgt:

$$\begin{aligned} m_{[i_1\ldots i_k][j_1\ldots j_k]} &= (\delta_{[i_1\ldots i_k][j_1\ldots j_k]} - z_{[i_1\ldots i_k][j_1\ldots j_k]} + z_{[j_1\ldots j_k][j_1\ldots j_k]})/q_{[j_1\ldots j_k]} \\ &= \Big(1 - p_{j_1j_2}\cdots p_{j_{k-1}j_k} \cdot \sum_l \left(\bar{z}_{i_kl} - q_l\right) p_{lj_1} \\ &\qquad + p_{j_1j_2}\cdots p_{j_{k-1}j_k} \cdot \sum_l \left(\bar{z}_{j_kl} - q_l\right) p_{lj_1}\Big) / (q_{j_1} \cdot p_{j_1j_2}\cdots p_{j_{k-1}j_k}) \\ &= \frac{1}{q_{j_1} \cdot p_{j_1j_2}\cdots p_{j_{k-1}j_k}} - \frac{\sum_l (\bar{z}_{i_kl} - \bar{z}_{j_kl}) p_{lj_1}}{q_{j_1}} \end{aligned}$$

Für die rückwärtslaufende Kette finden wir

$$\begin{aligned} \hat{p}_{[i_1\ldots i_k][j_1\ldots j_k]} &= \hat{p}_{i_1j_k} \cdot \hat{p}_{j_kj_{k-1}} \cdots \hat{p}_{j_2j_1} \\ &= \frac{q_{j_1}}{q_{i_1}} \cdot p_{j_1j_2}\cdots p_{j_{k-1}j_k} \cdot p_{j_ki_1} \\ \hat{p}^{(n)}_{[i_1\ldots i_k][j_1\ldots j_k]} &= \frac{q_{j_1}}{q_{i_1}} \cdot p_{j_1j_2}\cdots p_{j_{k-1}j_k} \cdot p_{j_ki_1}^{((n-1)k+1)} \end{aligned}$$

Damit folgt für die Fundamentalmatrix

$$\begin{aligned}
\hat{z}_{[i_1\ldots i_k][j_1\ldots j_k]} &= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + \sum_{n=1}^{\infty}(\hat{p}^{(n)}_{[i_1\ldots i_k][j_1\ldots j_k]} - q_{[j_1\ldots j_k]}) \\
&= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + p_{j_1 j_2}\cdots p_{j_{k-1}j_k}\cdot\frac{q_{j_1}}{q_{i_1}}\cdot\sum_{n=1}^{\infty}\Big(\sum_l \bar{p}^{(n-1)}_{j_k l}p_{li_1} - q_{i_1}\Big) \\
&= \delta_{[i_1\ldots i_k][j_1\ldots j_k]} + p_{j_1 j_2}\cdots p_{j_{k-1}j_k}\cdot\frac{q_{j_1}}{q_{i_1}}\cdot\sum_l\Big(\bar{z}_{j_k l} - q_l\Big)p_{li_1}
\end{aligned}$$

und für die Matrix der mittleren Erst-Eintritts-Zeiten

$$\begin{aligned}
\hat{m}_{[i_1\ldots i_k][j_1\ldots j_k]} &= (\delta_{[i_1\ldots i_k][j_1\ldots j_k]} - \hat{z}_{[i_1\ldots i_k][j_1\ldots j_k]} + \hat{z}_{[j_1\ldots j_k][j_1\ldots j_k]})/q_{[j_1\ldots j_k]} \\
&= \Big(1 - p_{j_1 j_2}\cdots p_{j_{k-1}j_k}\cdot\frac{q_{j_1}}{q_{i_1}}\cdot\sum_l\Big(\bar{z}_{j_k l} - q_l\Big)p_{li_1} \\
&\qquad + p_{j_1 j_2}\cdots p_{j_{k-1}j_k}\cdot 1\cdot\sum_l\Big(\bar{z}_{j_k l} - q_l\Big)p_{lj_1}\Big)/(q_{j_1}\cdot p_{j_1 j_2}\cdots p_{j_{k-1}j_k}) \\
&= \frac{1}{q_{j_1}\cdot p_{j_1 j_2}\cdots p_{j_{k-1}j_k}} - \sum_l \bar{z}_{j_k l}\Big(\frac{p_{li_1}}{q_{i_1}} - \frac{p_{lj_1}}{q_{j_1}}\Big)
\end{aligned}$$

3.4.2.2 Herleitung der erwarteten Suchzeit

Wir berechnen die asymptotische Wahrscheinlichkeit, daß ein Tupel $[j\ldots j]$ vor einem Tupel $[i\ldots i]$ auftritt, wenn man die expandierte Kette vom aktuellen Zustand $[l\ldots.]$ aus rückwärtsverfolgt. Da sich die Tupel nicht überlappen, genügt es, vom aktuellen Zustand nur das erste Element l zu kennen. Diese asymptotische Wahrscheinlichkeit sei mit $b([j\ldots j],[i\ldots i]\mid l)$ bezeichnet.

Lemma 3.17:

Es gilt

$$b([j\ldots j],[i\ldots i]\mid l) = \frac{A}{B}$$

mit dem Zähler

$$A = q_j p_{jj}^{k-1} \; - \; q_i q_j p_{ii}^{k-1} p_{jj}^{k-1}\cdot\sum_x\Big(\bar{z}_{jx} - \bar{z}_{ix}\Big)\Big(\frac{p_{xi}}{q_i} - \frac{p_{xl}}{q_l}\Big)$$

und dem Nenner

$$B = q_i p_{ii}^{k-1} \; + \; q_j p_{jj}^{k-1} \; - \; q_i q_j p_{ii}^{k-1} p_{jj}^{k-1}\cdot\sum_x\Big(\bar{z}_{jx} - \bar{z}_{ix}\Big)\Big(\frac{p_{xi}}{q_i} - \frac{p_{xj}}{q_j}\Big)$$

Beweis:
Wir setzen ein in Lemma 1.13 und erhalten

$$\begin{aligned}
& b([j\dots j],[i\dots i] \mid l) \\
&= \frac{\hat{m}_{[l\dots][i\dots i]} + \hat{m}_{[i\dots i][j\dots j]} - \hat{m}_{[l\dots][j\dots j]}}{\hat{m}_{[i\dots i][j\dots j]} + \hat{m}_{[j\dots j][i\dots i]}} \\
&= \frac{A'}{B'}
\end{aligned}$$

mit

$$\begin{aligned}
A' &= \frac{1}{q_i p_{ii}^{k-1}} - \sum_x \bar{z}_{ix}\left(\frac{p_{xl}}{q_l} - \frac{p_{xi}}{q_i}\right) - \sum_x \bar{z}_{jx}\left(\frac{p_{xi}}{q_i} - \frac{p_{xj}}{q_j}\right) + \sum_x \bar{z}_{jx}\left(\frac{p_{xl}}{q_l} - \frac{p_{xj}}{q_j}\right) \\
&= \frac{1}{q_i p_{ii}^{k-1}} - \sum_x \left(\bar{z}_{ix} - \bar{z}_{jx}\right)\left(\frac{p_{xl}}{q_l} - \frac{p_{xi}}{q_i}\right)
\end{aligned}$$

und

$$\begin{aligned}
B' &= \frac{1}{q_j p_{jj}^{k-1}} - \sum_x \bar{z}_{jx}\left(\frac{p_{xi}}{q_i} - \frac{p_{xj}}{q_j}\right) + \frac{1}{q_i p_{ii}^{k-1}} - \sum_x \bar{z}_{ix}\left(\frac{p_{xj}}{q_j} - \frac{p_{xi}}{q_i}\right) \\
&= \frac{1}{q_j p_{jj}^{k-1}} + \frac{1}{q_i p_{ii}^{k-1}} - \sum_x \left(\bar{z}_{jx} - \bar{z}_{ix}\right)\left(\frac{p_{xi}}{q_i} - \frac{p_{xj}}{q_j}\right)
\end{aligned}$$

Durch Erweitern mit $q_i q_j p_{ii}^{k-1} p_{jj}^{k-1}$ folgt das Ergebnis. ∎

Lemma 3.18:

Für die Batched-k-Nach-Vorne-Schieben-Regel gilt bei abhängigen Zugriffen

$$b'_k(j,i|i) = \frac{q_j p_{jj}^{k-1} \;-\; q_j p_{ii}^{k-1} p_{jj}^{k-1} \sum_x (\bar{z}_{jx} - \bar{z}_{ix})\left(p_{xi} \;-\; \sum_{n=1}^{k} p_{xi}^{(n)}/k\right)}{q_i p_{ii}^{k-1} \;+\; q_j p_{jj}^{k-1} \;-\; p_{ii}^{k-1} p_{jj}^{k-1} \sum_x (\bar{z}_{jx} - \bar{z}_{ix})(q_j p_{xi} - q_i p_{xj})} \quad (3.12)$$

Beweis:
Die asymptotische Wahrscheinlichkeit, daß j vor i steht beim Zugriff auf i, erhält man, indem man vom aktuellen Block ausgehend die Zugriffsfolge rückwärtsverfolgt. Im aktuellen Block kann der aktuelle Zugriff auf i an $1, 2, \dots, k$-ter Stelle stehen, jeweils mit Wahrscheinlichkeit $1/k$. Wenn er an n-ter Stelle steht, dann gehen wir $n-1$ Schritte rückwärts, um das erste Element des aktuellen Blocks zu erreichen.

$$\begin{aligned}
b'_k(j,i|i) &= \frac{1}{k} \sum_{n=1}^{k} \sum_l \hat{p}_{il}^{(n-1)} \cdot b([j\dots j],[i\dots i] \mid l) \\
&= \frac{1}{k q_i} \sum_{n=0}^{k-1} \sum_l q_l p_{li}^{(n)} \cdot b([j\dots j],[i\dots i] \mid l)
\end{aligned}$$

Den Nenner von $b([j\ldots j],[i\ldots i]\mid l)$ lassen wir stehen und rechnen die Summe im Zähler aus.

$$\begin{aligned}
&\frac{1}{kq_i}\sum_{n=0}^{k-1}\sum_l q_l p_{li}^{(n)}\Big(q_j p_{jj}^{k-1} - q_i q_j p_{ii}^{k-1} p_{jj}^{k-1}\sum_x(\bar z_{jx}-\bar z_{ix})(p_{xi}/q_i - p_{xl}/q_l)\Big)\\
=\;& \frac{q_j p_{jj}^{k-1}}{kq_i}\sum_{n=0}^{k-1}\sum_l q_l p_{li}^{(n)}\\
&\qquad - q_j p_{ii}^{k-1} p_{jj}^{k-1}\sum_{n=0}^{k-1}\sum_x\Big(\bar z_{jx}-\bar z_{ix}\Big)\Big(\sum_l p_{xi}\frac{q_l p_{li}^{(n)}}{q_i} - \sum_l p_{xl}p_{li}^{(n)}\Big)/k\\
=\;& q_j p_{jj}^{k-1} \;-\; q_j p_{ii}^{k-1} p_{jj}^{k-1}\sum_{n=0}^{k-1}\sum_x(\bar z_{jx}-\bar z_{ix})\Big(p_{xi} - p_{xi}^{(n+1)}\Big)/k\\
=\;& q_j p_{jj}^{k-1} \;-\; q_j p_{ii}^{k-1} p_{jj}^{k-1}\sum_x(\bar z_{jx}-\bar z_{ix})\Big(kp_{xi} - \sum_{n=1}^{k} p_{xi}^{(n)}\Big)/k
\end{aligned}$$

∎

Satz 3.19:

Die erwartete Suchzeit für die Batched-k-Nach-Vorne-Schieben-Regel bei einer Markov-Zugriffsfolge ist

$$\mu = 1 + \sum_{i\in S}\sum_{j\neq i} q_i \cdot b'_k(j,i|i)$$

wobei $b'_k(j,i|i)$ gemäß Lemma 3.18 gegeben ist.

Bei unabhängigen Zugriffen erhält man wieder das Ergebnis von Gonnet/Munro/Suwanda (siehe Lemma 3.3).

Korollar 3.20:

Bei unabhängigen Zugriffen gilt für die Formeln (3.2) und (3.12) und für beliebige k die Gleicheit für alle $i\neq j$.

$$b'_k(j,i) = b'_k(j,i|i)$$

Beweis:
Bei unabhängigen Zugriffen ist $p_{ij}=q_j$ und $p_{ij}^{(n)}=q_j$. Übrig bleibt

$$b'_k(j,i|i) \;=\; \frac{q_j p_{jj}^{k-1}}{q_i p_{ii}^{k-1} + q_j p_{jj}^{k-1}}$$

$$
\begin{aligned}
&= \frac{q_j^k}{q_i^k + q_j^k} \\
&= b'_k(j,i)
\end{aligned}
$$

■

3.4.2.3 Beispiel

Wir betrachten wieder das Standardbeispiel von Seite 20. Es ist

$$p_{ij} = \left\{ \begin{array}{ccc} \alpha & : & i = j \\ \beta & : & i \neq j \end{array} \right\}, \tag{3.13}$$

außerdem $q_j = 1/n$ sowie

$$
\begin{aligned}
\bar{z}_{ij} &= \delta_{ij} + (\delta_{ij} - q_j)\frac{(n\alpha - 1)^k}{(n-1)^k - (n\alpha - 1)^k} \\
&= \left\{ \begin{array}{ccc} 1 + \dfrac{n-1}{n}\dfrac{(n\alpha-1)^k}{(n-1)^k - (n\alpha-1)^k} & : & i = j \\ -\dfrac{1}{n}\dfrac{(n\alpha-1)^k}{(n-1)^k - (n\alpha-1)^k} & : & i \neq j \end{array} \right\},
\end{aligned}
$$

was sich aus

$$
\begin{aligned}
p_{ij}^{(m)} &= \frac{1}{n} + \left(\delta_{ij} - \frac{1}{n}\right)\left(\frac{n\alpha - 1}{n-1}\right)^m \\
z_{ij} &= \delta_{ij} + \left(\delta_{ij} - \frac{1}{n}\right)\sum_{m=1}^{\infty}\left(\frac{n\alpha - 1}{n-1}\right)^m
\end{aligned}
$$

(vgl. Lemma 1.19) ergibt. Wir setzen ein in (3.12) und finden für den Zähler von $b'_k(j,i|i)$

$$\frac{\alpha^{k-1}}{n} \cdot \left(1 \; + \; \alpha^{k-1}\frac{n\alpha - 1}{n-1}\,\frac{(n-1)^k}{(n-1)^k - (n\alpha-1)^k} \; - \; \frac{\alpha^{k-1}}{k}\,\frac{n\alpha - 1}{n - n\alpha}\right),$$

für den Nenner

$$2\,\frac{\alpha^{k-1}}{n}\left(1 \; + \; \alpha^{k-1}\frac{n\alpha - 1}{n-1}\,\frac{(n-1)^k}{(n-1)^k - (n\alpha-1)^k}\right),$$

zusammen also

$$b'_k(j,i|i) = \frac{1}{2} - \frac{\alpha^{k-1}}{2k}\,\frac{n\alpha-1}{n-n\alpha}\,\frac{(n-1)^k-(n\alpha-1)^k}{(n-1)^k-(n\alpha-1)^k+\alpha^{k-1}(n\alpha-1)(n-1)^{k-1}}$$

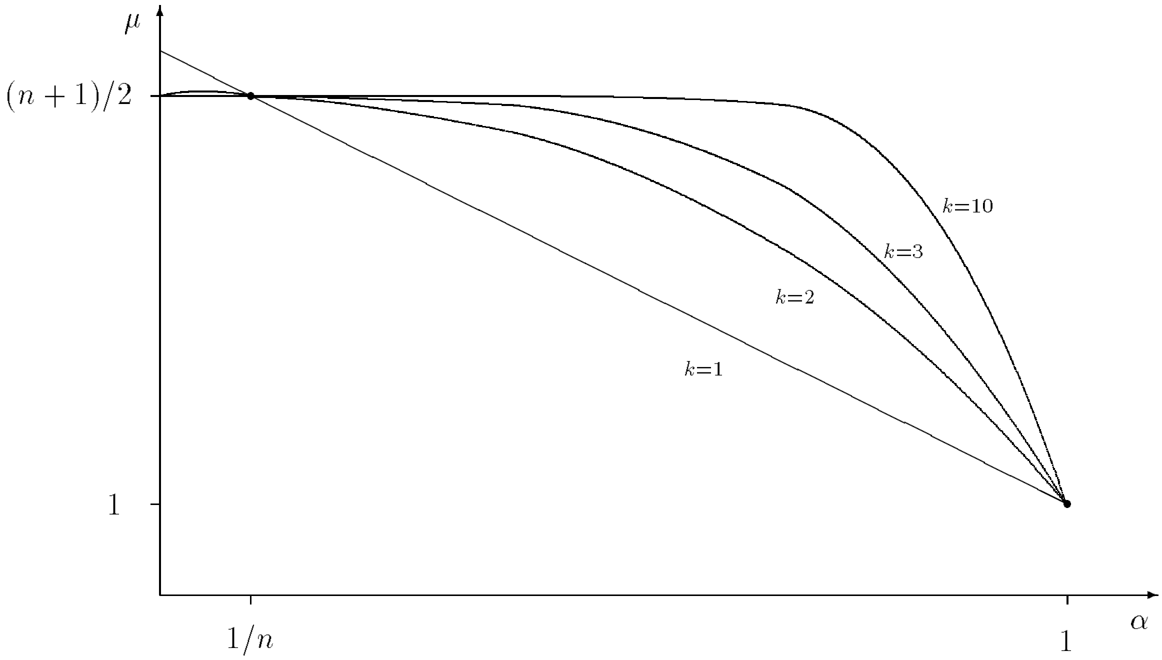
Damit kann man die erwartete Suchzeit ausrechnen.

Satz 3.21:

Für die erwartete Suchzeit der Batched-k-Nach-Vorne-Schieben-Regel bei der Zugriffsfolge (3.13) gilt

$$\mu = 1 + \frac{n-1}{2} - \frac{n-1}{2}\,\frac{\alpha^{k-1}}{k}\,\frac{n\alpha-1}{n-n\alpha}\,\frac{(n-1)^k-(n\alpha-1)^k}{(n-1)^k-(n\alpha-1)^k+\alpha^{k-1}(n\alpha-1)(n-1)^{k-1}}$$

Folgendes Diagramm veranschaulicht die Suchzeit in Abhängigkeit von α für verschiedene k.



Für $k = 1$ erhält man wieder $\mu = 1 + (1-\alpha)n/2$.

Da bei der Batched-k-Regel die Heuristik noch seltener aufgerufen wird als bei der Simple-k-Regel, ist die Anpassung bei trägen Übergängen schlechter. Für die Zugriffsfolge des Beispiels sieht man dies leicht.

Satz 3.22:

Bei trägem Übergangsverhalten ($\alpha > 1/n$) der Zugriffsfolge (3.13) gilt, daß für $n \geq 2$ und jedes $k \geq 2$ die erwartete Suchzeit unter der Simple-k-Nach-Vorne-Schieben-Regel besser ist als unter der Batched-k-Nach-Vorne-Schieben-Regel.

Beweis:
Im Fall $n = 1$ sind natürlich alle erwarteten Suchzeiten gleich 1.
Wir zeigen, daß für $n \geq 2$, $\alpha > 1/n$, $k \geq 2$ und alle $i \neq j$ gilt

$$b_k(j, i|i) < b'_k(j, i|i),$$

woraus die Behauptung folgt.

$$\begin{aligned}
&b'_k(j, i|i) - b_k(j, i|i) \\
&= \frac{1}{2} - \frac{\alpha^{k-1}}{2k}\frac{n\alpha - 1}{n - n\alpha}\frac{(n-1)^k - (n\alpha - 1)^k}{(n-1)^k - (n\alpha-1)^k + \alpha^{k-1}(n\alpha - 1)(n-1)^{k-1}} - \frac{n(1-\alpha^k)}{2(n-\alpha^{k-1})}
\end{aligned}$$

Der Hauptnenner ist für $1/n < \alpha < 1$ positiv, und der Zähler läßt sich als Produkt schreiben

$$\begin{aligned}
&\alpha^{k-1}(n\alpha - 1) \cdot D(\alpha) \qquad \text{mit} \\
&D(\alpha) = \Big(\big(kn(1-\alpha) - n + \alpha^{k-1}\big) \cdot \big((n-1)^k - (n\alpha - 1)^k\big) \\
&\qquad\qquad + kn(1-\alpha)(n\alpha - 1)\alpha^{k-1}(n-1)^{k-1}\Big)
\end{aligned}$$

Der erste Faktor $\alpha^{k-1}(n\alpha - 1)$ ist im betrachteten Bereich positiv. Für den zweiten Faktor $D(\alpha)$ machen wir die Substitution $x = 1 - \alpha$; der betrachtete Bereich ist also $0 < x < 1 - 1/n$.
Wir benutzen die Bernoulli'sche Ungleichung

$$a^n \geq 1 + n(a-1) \ \text{ für } \ a \geq 0$$

um $(1-x)^{k-1}$ abzuschätzen. Das ist möglich wegen $(1 - x) > 0$.

$$\begin{aligned}
F(x) = D(1-\alpha) &= \big(knx - n + (1-x)^{k-1}\big) \cdot \big((n-1)^k - (n-1-nx)^k\big) \\
&\quad + knx(n-1-nx)(1-x)^{k-1}(n-1)^{k-1} \\
&\geq \big(knx - n + 1 - (k-1)x\big) \cdot \big((n-1)^k - (n-1-nx)^k\big) \\
&\quad + knx(n-1-nx)(1-x)^{k-1}(n-1)^{k-1}
\end{aligned}$$

1. Fall: $knx - n + 1 - (k-1)x \geq 0 \iff x \geq (n-1)/(kn-k+1)$
   In diesem Fall folgt sofort $F(x) > 0$.

2. Fall: $knx - n + 1 - (k-1)x < 0 \iff x < (n-1)/(kn-k+1)$
   Wir wenden die Bernoulli'sche Ungleichung noch zweimal an.

$$\begin{aligned}
F(x) &\geq \Big(knx - n + 1 - (k-1)x\Big) \cdot \Big((n-1)^k - (n-1)^k(1 - \frac{n}{n-1}\, x)^k\Big) \\
&\quad + knx(n-1-nx)(1-(k-1)x)(n-1)^{k-1} \\
&\geq \Big(knx - n + 1 - (k-1)x\Big) \cdot (n-1)^k \cdot \Big(1 - 1 + k\,\frac{n}{n-1}\, x\Big) \\
&\quad + knx(n-1-nx)(1-(k-1)x)(n-1)^{k-1} \\
&= (n-1)^{k-1}\Big((knx - n + 1 - (k-1)x) \cdot knx \\
&\quad + knx(n-1-nx)(1-(k-1)x)\Big) \\
&= (n-1)^{k-1}(k-1)kn^2x^3 \\
&> 0
\end{aligned}$$

   für $x > 0$.

■

Bei unabhängigen Zugriffen gilt, daß für jedes $k > 1$ die Batched-k-Regel besser ist als die Simple-k-Regel (siehe Satz 3.2). Bei abhängigen Zugriffen mit trägem Übergangsverhalten finden wir die Umkehrung: die Simple-k-Regel ist besser als die Batched-k-Regel.

3.5 Zusammenfassung

In diesem Kapitel haben wir das Verhalten der k-Mal-Hintereinander-Regeln bei abhängigen Zugriffen untersucht.
Diese Regeln zeigen bei **unabhängigen Zugriffen** eine gute erwartete Suchzeit, die für wachsende k immer besser wird. Ein einzelner Zugriff auf ein selten nachgefragtes Element zerstört nicht die bisher erreichte Approximation an die optimale Ordnung. Für steigende k nimmt die Wahrscheinlichkeit des k-maligen Zugriffs (p^k) auf selten nachgefragte Elemente stärker ab als sie dies bei häufig nachgefragten Elementen macht.
Bei Anfragefolgen, die durch Markov-Ketten **mit Lokalität** beschrieben werden, wird die erwartete Suchzeit für wachsende k immer schlechter. Dies macht sich deutlich bei elementweiser Lokalität bemerkbar; bei einem Wechsel von einem Element zu einem anderen wird die Reaktion der Heuristik um k Schritte verzögert. Wir fassen diese Betrachtung zusammen in folgender Bemerkung:

Bemerkung 3.23:

> *Bei unabhängigen Zugriffen wird eine einmal erreichte gute Anordnung der Liste bei steigenden k immer weniger gestört.*
> *Bei abhängigen Zugriffen mit Lokalität wird die Anpassung an eine neue Situation bei steigenden k immer weiter verzögert.*

Wir verallgemeinern dies zu folgender Interpretation. Diese Deutung hat keine allgemeine Gültigkeit, denn sonst wäre etwa die Heuristik, die gar keine Umordnung vornimmt, bei unabhängigen Zugriffen optimal. Vielmehr soll sie eine Tendenz zum Vergleich von Regeln beschreiben.

Interpretation 3.24:

> *Bei unabhängigen Zugriffen ist eine Selbst-Organisations-Regel asymptotisch umso besser, je langsamer sie die Änderungen vornimmt.*
> *Bei abhängigen Zugriffen mit Lokalität ist eine solche Regel umso besser, je schneller sie sich an neue Situationen anpassen kann.*

Diese Sicht wird gestützt von der Tatsache, daß die Nach-Vorne-Schieben-Regel für gewisse Klassen von abhängigen Zugriffsfolgen optimal ist [9]. In der Tat ist die Nach-Vorne-Schieben-Regel diejenige Heuristik, die die Umordnung der Liste am schnellsten und radikalsten vornimmt.

Für die Vertauschen-Regel gilt, daß ihre erwartete Suchzeit bei unabhängigen, nicht-trivialen Zugriffen stets besser ist als die der Nach-Vorne-Schieben-Regel [35].
Angesichts der oben gemachten Interpretation vermuten wir, daß bei abhängigen Zugriffen mit Lokalität die Vertauschen-Regel deutlich schlechter ist als die Nach-Vorne-Schieben-Regel.

Die Untersuchung dieser Vermutung für einen Spezialfall ist Gegenstand des nächsten Kapitels.

Kapitel 4

Die Vertauschen-Regel

In Kapitel 1 wurde die Nach-Vorne-Schieben-Regel bei abhängigen Zugriffen betrachtet. In diesem Kapitel sollen Ansätze aufgezeigt werden, um weitere Heuristiken für lineare Listen bei abhängigen Zugriffen untersuchen zu können. Insbesondere wird auf die Vertauschen-Regel eingegangen.
Anstatt wie in den ersten drei Kapiteln eine Rückwärtsanalyse durchzuführen, betrachten wir jetzt direkt die Zustände der Datenstruktur und ihre stationäre Verteilung.

4.1 Unabhängige Zugriffe

Die allgemeingültigste Aussage bei unabhängigen, identisch verteilten Zugriffen ist die von Rivest [35], daß die Vertauschen-Regel bei allen Verteilungen außer der Gleichverteilung stets bessere Suchkosten aufweist als die Nach-Vorne-Schieben-Regel (siehe Satz 1.5).
Um dies zeigen zu können, muß man die stationäre Verteilung der zugrundeliegenden Markov-Kette betrachten. Jede Permutation $(i_1, \ldots, i_n)$ der Listenelemente bildet einen Zustand. Ein Übergang existiert, wenn durch Anwenden der Vertauschen-Regel eine Konfiguration der Liste aus einer anderen entsteht.
Sei $q(i_1, \ldots, i_n)$ die stationäre Wahrscheinlichkeit, daß sich die Liste in Konfiguration $(i_1, \ldots, i_n)$ befindet. Das Gleichungssystem für die stationäre Verteilung (*steady-state equations*) lautet:

$$\begin{aligned} q(i_1, \ldots, i_n) &= p_{i_1} \cdot q(i_1, \ldots, i_n) \\ &\quad + \sum_{k=1}^{n-1} p_{i_k} \cdot q(i_1, \ldots, i_{k+1}, i_k, \ldots, i_n) \end{aligned} \tag{4.1}$$

Mehrere Autoren, die die Vertauschen-Regel und weitere Regeln betrachtet haben, beschränken sich auf die spezielle Zugriffsverteilung (siehe [21, 25, 33, 36])

$$\begin{aligned} p_1 &= \alpha \\ p_i &= \beta = (1-\alpha)/(n-1) \text{ für } i = 2, \ldots, n \end{aligned} \tag{4.2}$$

Hier konnte gezeigt werden, daß die Vertauschen-Regel optimal ist und die Nach-Vorne-Schieben-Regel am schlechtesten ist innerhalb aller Vorwärts-Regeln. Außerdem konnten für verschiedene Regel-Spektren Hierarchien aufgestellt werden, beispielsweise für die k-Plätze-Vorwärts-Regeln für $k = 1, \ldots, n-2$

$$\mu(\text{MOVE-UP}(k)) \leq \mu(\text{MOVE-UP}(k+1)).$$

Die Resultate sind in Satz 1.7 auf Seite 11 zusammengefaßt.

Die Idee zur Verwendung der Verteilung (4.2) ist, daß man die $n!$ Zustände der zugrundeliegenden Markov-Kette reduzieren kann auf n Zustände: da nur Element 1 ausgezeichnet ist, genügt es, nur seine Position zu wissen.
Sei $q(j)$ die stationäre Wahrscheinlichkeit, daß sich Element 1 auf Position j befindet. Dann ist die Festsetzung

$$q(j) = q(i_1, \ldots, i_{j-1}, 1, i_{j+1}, \ldots, i_n)$$

wohldefiniert, wenn $i_1, \ldots, i_{j-1}, i_{j+1}, \ldots, i_n$ alle Permutationen von $\{2, \ldots, n\}$ durchlaufen.

Genauergesagt definiert man eine Partition $\{X_j\}$ des Zustandsraums, hier also

$$\begin{aligned} X_j \;=\; \Big\{ & (i_1, \ldots, i_{j-1}, 1, i_{j+1}, \ldots, i_n) : \\ & (i_1, \ldots, i_{j-1}, i_{j+1}, \ldots, i_n) \text{ ist Permutation von } 2, \ldots, n \Big\} \end{aligned}$$

Wenn für alle Teilmengen X_j, X_k gilt, daß für jedes Element aus X_j die Übergangswahrscheinlichkeiten zu einem beliebigen Element in X_k gleich sind, dann heißt die Markov-Kette *partitionierbar* (*lumpable*) bezüglich der Partition $\{X_j\}$.

In diesem Fall kann man eine neue Markov-Kette definieren, deren Zustände aus den Teilmengen X_j bestehen. Die Übergangswahrscheinlichkeiten $p_{X_j X_k} = p_{il}$ sind wohldefiniert für beliebige $i \in X_j$, $l \in X_k$. Die stationäre Wahrscheinlichkeit des Zustands X_j der neuen Kette ist dann $q(X_j) = \sum_{i \in X_j} q_i$.
Oben haben wir $q(j) = q_i$ ($i \in X_j$ beliebig), also $q(j) = q(X_j)/(n-1)!$ setzen können, da die stationären Wahrscheinlichkeiten innerhalb von X_j gleich sind.

4.2 Abhängige Zugriffe

Bei Zugriffen in Form einer Markov-Kette sind empirische Ergebnisse von Lam/Leung/Siu [26] bekannt. Sie haben Experimente durchgeführt, die gezeigt haben, daß die erwartete Suchzeit bei der Vertauschen-Regel sowohl besser als auch schlechter als bei der Nach-Vorne-Schieben-Regel sein kann.

Um eine Analyse in gewissen Fällen zu ermöglichen, wollen wir die Idee von oben übertragen und auch bei abhängigen Zugriffen die Anzahl der Zustände der zugrundeliegenden Markov-Kette reduzieren. Die betrachtete Markov-Kette ist jetzt die erweiterte Kette mit den Zuständen

$$(i_1, \ldots, i_n \mid k) \in \mathcal{S}_n \times \{1, \ldots, n\}.$$

Dieser Zustand bedeutet, daß $(i_1, \ldots, i_n)$ die aktuelle Konfiguration der Liste ist, und daß k das Element ist, das als nächstes nachgefragt wird.
Die Gleichungen für die stationäre Verteilung unter der Vertauschen-Regel lauten dann:

$$\begin{aligned} q(i_1, \ldots, i_n \mid i_j) &= p_{i_1 i_j} \cdot q(i_1, \ldots, i_n \mid i_1) \\ &\quad + \sum_{k=1}^{n-1} p_{i_k i_j} \cdot q(i_1, \ldots, i_{k+1}, i_k, \ldots, i_n \mid i_k) \end{aligned} \tag{4.3}$$

Welche Zugriffsfolgen sind geeignet, um den Zustandsraum zu reduzieren?

Die Markov-Kette aus Beispiel 1 (Seite 20) ist gegeben durch die Übergangsmatrix $P = (p_{ij})$ mit

$$p_{ij} = \left\{ \begin{array}{ccc} \alpha & : & i = j \\ \beta & : & i \neq j \end{array} \right\}. \tag{4.4}$$

Wichtig für den nächsten Übergang ist also nur das zuletzt nachgefragte Element. Ein Zustand $(i_1, \ldots, i_n \,|\, i_j)$ der erweiterten Kette beschreibt die Situation, daß der nächste Zugriff auf das Element auf Position j erfolgen wird. Der Zustandsraum der Größe $nn!$ kann verkleinert werden auf die Größe n, indem der Zustand nur die Position des Elements angibt, das als nächstes nachgefragt wird.
Für den Rest des Kapitels werden wir nur noch diese Zugriffsfolge betrachten.

Sei $q(j)$ die stationäre Wahrscheinlichkeit, daß sich das Element, was als nächstes nachgefragt wird, auf Position j befindet. Wir setzen

$$q(j) = q(i_1, \ldots, i_n \mid i_j) \tag{4.5}$$

Lemma 4.1:

> *Für die Zugriffsfolge (4.4) ist die Festsetzung (4.5) der stationären Wahrscheinlichkeiten wohldefiniert und besitzt eine eindeutige Lösung.*

Beweis:
Wir zeigen, daß alle $nn!$ Gleichungen des stationären Systems (4.3) erfüllt werden, und berechnen die Lösung des Systems.

1. FALL: $j = 1$

$$\begin{aligned}
q(1) &= q(i_1, \ldots, i_n \mid i_1) \\
&= \alpha q(i_1, \ldots, i_n | i_1) + \alpha q(i_2, i_1, i_3, \ldots, i_n | i_1) \\
&\quad + \sum_{k=2}^{n-1} \beta q(i_1, \ldots, i_{k+1}, i_k, \ldots, i_n | i_k) \\
&= \alpha q(1) + \alpha q(2) + \beta \sum_{k=2}^{n-1} q(k+1) \\
&= (\alpha - \beta)(q(1) + q(2)) + \beta \sum_{k=1}^{n} q(k)
\end{aligned}$$

2. FALL: $j = 2, \ldots, n-1$

$$\begin{aligned}
q(j) &= q(i_1, \ldots, i_n \mid i_j) \\
&= \beta q(i_1, \ldots, i_n | i_1) + \sum_{\substack{k=1 \\ k \neq j}}^{n-1} \beta q(i_1, \ldots, i_{k+1}, i_k, \ldots, i_n | i_k) \\
&\quad + \alpha q(i_1, \ldots i_{j+1}, i_j, \ldots, i_n | i_j) \\
&= (\alpha - \beta) q(j+1) + \beta \sum_{k=1}^{n} q(k)
\end{aligned}$$

3. FALL: $j = n$

$$\begin{aligned}
q(n) &= q(i_1, \ldots, i_n \mid i_n) \\
&= \beta q(i_1, \ldots, i_n | i_1) + \sum_{k=1}^{n-1} \beta q(i_1, \ldots, i_{k+1}, i_k, \ldots, i_n | i_k) \\
&= \beta \sum_{k=1}^{n} q(k)
\end{aligned}$$

Das ursprüngliche System läßt sich also auf n Gleichungen reduzieren. Zur eindeutigen Lösung kommt hinzu, daß die Summe über alle Wahrscheinlichkeiten gerade 1 ist:

$$\sum_{\pi} \sum_{k} q(\pi \mid k) = 1$$

$$\sum_{k} q(k) = 1/n!$$

Aus Fall 3 folgt sofort:

$$q(n) = \beta / n!$$

Durch sukzessives Einsetzen erhält man $q(n-1), q(n-2), \ldots$:

$$\begin{aligned}
q(j) &= \frac{\beta}{n!} \cdot \frac{1 - (\alpha - \beta)^{n-j+1}}{1 - \alpha + \beta} \quad \text{für } j = 2, \ldots, n-1, \\
q(1) &= \frac{\beta}{n!} \cdot \frac{1 - (\alpha - \beta)^{n}}{(1 - \alpha + \beta)^2}
\end{aligned}$$

■

Satz 4.2:

Für die erwarteten Suchkosten der Vertauschen-Regel bei der Zugriffsfolge gemäß (4.4) gilt:

$$\begin{aligned}\mu \;=\; & \frac{\beta}{2(1-\alpha+\beta)^3}\cdot\Big(2(\alpha-\beta)^{n-2}\cdot\big(2\beta^3+\beta^2-6\alpha\beta^2-2\alpha\beta+6\alpha^2\beta+\alpha^2-2\alpha^3\big)\\ & \quad -2(\alpha-\beta)^n(1-\alpha+\beta)+\beta^2(n^2+3n)+\beta(2n^2+4n-2)\\ & \quad -\alpha\beta(2n^2+6n)-\alpha(2n^2+4n-2)+\alpha^2(n^2+3n)+(n^2+n)\Big)\end{aligned}$$

Beweis:
Die Suchzeit in Konfiguration j ist gerade j. Wir bilden den Erwartungswert:

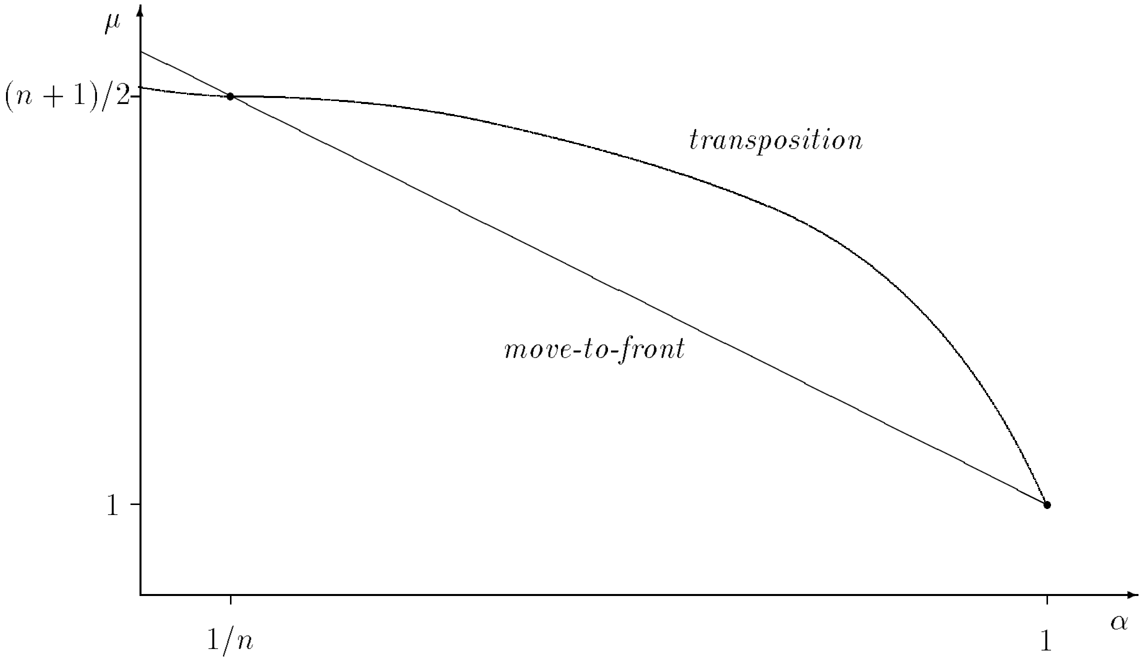
$$\begin{aligned}\mu &= \sum_{(\pi|j)} q(\pi|\,j)\cdot j\\ &= n!\cdot\sum_j q(j)\cdot j\\ &= n!\cdot\frac{\beta}{n!}\cdot\Big(\frac{1-(\alpha-\beta)^n}{(1-\alpha+\beta)^2}+\sum_{j=2}^{n}\frac{1-(\alpha-\beta)^{n-j+1}}{1-\alpha+\beta}\cdot j\Big)\\ &= \frac{\beta}{(1-\alpha+\beta)^2}\cdot\Big(1-(\alpha-\beta)^n+(1-\alpha+\beta)\sum_{j=2}^{n}(1-(\alpha-\beta)^{n-j+1})\cdot j\Big)\end{aligned}$$

Wenn man die Summe ausrechnet, kommt man auf obigen Ausdruck, unter Benutzung von

$$\sum_{j=1}^{n} j\cdot q^j \;=\; \frac{q+q^{n+1}(nq-n-1)}{(q-1)^2}$$

■

Folgendes Diagramm vergleicht die Nach-Vorne-Schieben-Regel (*move-to-front*) und die Vertauschen-Regel (*transposition*).



4.3 Vergleich weiterer Regeln

Wir wollen das Ergebnis des letzten Abschnitts verallgemeinern und die k-Plätze-Vorwärts-Regeln (MOVE-UP(k)) analysieren. Die Vertauschen-Regel ($k = 1$) und die Nach-Vorne-Schieben-Regel ($k = n-1$) sind darin als Spezialfälle enthalten.
Das Gleichungssystem der stationären Verteilung lautet:

$$\begin{aligned} q(i_1,\ldots,i_n \mid i_j) &= p_{i_1 i_j} q(i_1,\ldots,i_n \mid i_1) \\ &\quad + \sum_{l=2}^{k+1} p_{i_1 i_j} q(i_2,\ldots,i_l,i_1,i_{l+1},\ldots,i_n \mid i_1) \\ &\quad + \sum_{l=2}^{n-k} p_{i_l i_j} q(i_1,\ldots,i_{l-1},i_{l+1},\ldots,i_{l+k},i_l,i_{l+k+1},\ldots,i_n \mid i_l) \end{aligned} \tag{4.6}$$

Wir führen wieder die Reduktion durch

$$q(j) = q(i_1,\ldots,i_n \mid i_j).$$

Für die stationäre Wahrscheinlichkeit $q(j)$, daß auf das j-te Listenelement zugegriffen wird, erhält man aus dem System (4.6):

1. Fall: $j = 1$

$$\begin{aligned} q(1) &= \alpha q(1) + \sum_{l=2}^{k+1} \alpha q(l) + \sum_{l=2}^{n-k} \beta q(l+k) \\ &= \alpha \cdot \sum_{l=1}^{k+1} q(l) \ + \ \beta \cdot \sum_{l=k+2}^{n} q(l) \end{aligned}$$

2. Fall: $j = 2, \ldots, n-k$

$$\begin{aligned} q(j) &= \beta q(1) + \sum_{l=2}^{k+1} \beta q(l) + \sum_{l=2}^{n-k} \beta q(l+k) + (\alpha - \beta) q(j+k) \\ &= (\alpha - \beta) q(j+k) + \beta \cdot \sum_{l=1}^{n} q(l) \end{aligned}$$

3. Fall: $j = n-k+1, \ldots, n$

$$\begin{aligned} q(j) &= \beta q(1) + \sum_{l=2}^{k+1} \beta q(l) + \sum_{l=2}^{n-k} \beta q(l+k) \\ &= \beta \cdot \sum_{l=1}^{n} q(l) \end{aligned}$$

Wegen $\sum_j q(j) = 1/n!$ folgt für Fall 3:

$$q(j) = \frac{\beta}{n!} \quad \text{für} \quad j = n-k+1, \ldots, n.$$

In Fall 2 gilt für $n+1-(t+1)k \leq j \leq n-tk$, $t = 1, \ldots, \lfloor (n-2)/k \rfloor$, und $j \geq 2$ im Fall $t = \lfloor (n-2)/k \rfloor$

$$q(j) = \frac{\beta}{n!} \frac{1 - (\alpha - \beta)^{t+1}}{1 - \alpha + \beta}, \quad \text{mit} \quad t = \left\lfloor \frac{n-j}{k} \right\rfloor \tag{4.7}$$

Von n angefangen rückwärts ist für jeweils k Werte von j der Exponent t gleich. Der Fall 3 ist mit $t = 0$ in obiger Formel enthalten.
Den Fall 1 ($j = 1$) stellen wir dar durch

$$q(1) = \frac{1}{n!} - \sum_{j=2}^{n} q(j)$$

Satz 4.3:

Im Fall träger Übergänge ($\alpha \geq 1/n$) gilt für die erwarteten Suchzeiten bei $k = 1, \ldots, n-2$:

$$\mu(\text{MOVE-UP}(k+1)) \leq \mu(\text{MOVE-UP}(k)) \tag{4.8}$$

Beweis:
Sei $q_k(j)$ die stationäre Wahrscheinlichkeit des Zustands j bei der k-Plätze-Vorwärts-Regel. Dann ist

$$\mu(\text{MOVE-UP}(k)) = \sum_{j=1}^{n} q_k(j) \cdot j$$

Für die Differenz gilt

$$\begin{aligned}
&\mu(\text{MOVE-UP}(k)) - \mu(\text{MOVE-UP}(k+1)) \\
&= \sum_{j=1}^{n} \Big(q_k(j) - q_{k+1}(j)\Big) \cdot j \\
&= \Big(q_k(1) - q_{k+1}(1)\Big) + \sum_{j=2}^{n} \Big(q_k(j) - q_{k+1}(j)\Big) \cdot j \\
&= \sum_{j=2}^{n} \Big(q_{k+1}(j) - q_k(j)\Big) + \sum_{j=2}^{n} \Big(q_k(j) - q_{k+1}(j)\Big) \cdot j \\
&= \sum_{j=2}^{n} \Big(q_k(j) - q_{k+1}(j)\Big) \cdot (j-1) \\
&= \frac{\beta}{n!(1-\alpha+\beta)} \sum_{j=2}^{n} \Big((\alpha-\beta)^{\lfloor\frac{n-j}{k+1}\rfloor+1} - (\alpha-\beta)^{\lfloor\frac{n-j}{k}\rfloor+1}\Big) \cdot (j-1) \\
&\geq 0
\end{aligned}$$

da bei trägen Übergängen $0 < (\alpha - \beta) < 1$ ist, und für alle $j = 2, \ldots, n$ gilt

$$\left\lfloor \frac{n-j}{k+1} \right\rfloor \leq \left\lfloor \frac{n-j}{k} \right\rfloor.$$

■

Bei trägen Übergängen gilt also, daß die Hierarchie der k-Plätze-Vorwärts-Regeln bzgl. der erwarteten Suchzeit gerade umgekehrt ist als bei unabhängigen Zugriffen.

4.4 Zusammenfassung und Ausblick

In diesem Kapitel haben wir die Idee verfolgt, durch Beschränkung auf spezielle Zugriffsfolgen die Größe des Zustandsraums zu reduzieren und dadurch die Analyse selbst-organisierender Datenstrukturen zu ermöglichen.

Mehrere Autoren haben diesen Ansatz bei unabhängigen Zugriffen angewendet, und konnten innerhalb gewisser Klassen von Regeln Hierarchien aufstellen.
Bei abhängigen Zugriffen können wir mit Hilfe dieser Idee eine Hierarchie innerhalb der k-Plätze-Vorwärts-Regeln zeigen. Insbesondere ist im Fall $k = 1$ auch die Analyse der Vertauschen-Regel eingeschlossen.
Bei trägem Übergangsverhalten stellt sich heraus, daß die Rangfolge der Regeln (bzgl. der erwarteten Suchzeit) gerade umgekehrt ist als bei unabhängigen Zugriffen. Das entspricht der in (3.24) gemachten Interpretation. Wir vermuten, daß für die anderen Regel-Spektren POS(k) und SWITCH(k) ebenfalls diese Umkehrung gilt. Es wäre interessant, dies allgemein für die in (Lam [25]) definierte Ordnung von Regeln zu zeigen.

Weiter wäre es wünschenswert, ein Kriterium oder sogar ein Maß für die Trägheit der Übergänge zu finden.
Wir haben Beispiele betrachtet, denen das Phänomen der Lokalität leicht anzusehen ist. Außerdem ist bei diesen Beispielen die Lokalität durch einen Parameter regelbar. Den Parameter kann man als Maß für die Trägheit der Übergänge auffassen, und wir konnten die erwartete Suchzeit in Abhängigkeit von diesem Parameter bestimmen.
Wenn man für jede Markov-Kette ein Maß für die Trägheit der Übergänge finden könnte, wäre es möglich, an der Zugriffsfolge direkt abzulesen, wie gut das Verhalten selbst-organisierender Datenstrukturen sein wird.

Anhang A

Markov-Ketten

In diesem Abschnitt werden Grundlagen aus der Theorie der Markov-Ketten zusammengefaßt. Die Darstellung beruht auf Kemeny/Snell [22] sowie auf Chung [13] und Feller [14].

A.1 Grundlagen

Eine Markov-Kette ist eine Folge $\{X_m\}_{m\geq 0}$ von Zufallsvariablen mit gemeinsamem Wertebereich $S = \{1, 2, \ldots, n\}$. Durch die Übergangsmatrix $P = (p_{ij})_{i,j\in S}$ und die Startverteilung $a = (a_i)_{i\in S}$ ist die Markov-Kette vollständig definiert. Dabei ist

$$p_{ij} = p_{ij}^{(1)} = Prob(X_{m+1} = j \mid X_m = i)$$

die bedingte Wahrscheinlichkeit, in einem Schritt nach j zu kommen, wenn sich die Kette in Zustand i befindet. Die Wahrscheinlichkeitsverteilung der Variablen X_{m+1} ist also abhängig von der Realisierung der direkt vorausgehenden Variablen X_m. Die charakteristische Eigenschaft von Markov-Ketten (erster Ordnung) ist, daß diese Verteilung ausschließlich von der Realisierung von X_m abhängt, unabhängig von den Realisierungen vorhergehender Variablen $X_{m-1}, X_{m-2}, \ldots$.
Man kann die Übergangswahrscheinlichkeiten iterieren und erhält mit vollständiger Induktion:

$$p_{ij}^{(s+1)} = \sum_{k\in S} p_{ik} p_{kj}^{(s)}$$

die Wahrscheinlichkeit, in genau $s+1$ Schritten von i nach j zu gelangen. Also gilt für die Übergangsmatrix bei genau s Schritten:

$$P^{(s)} = P^s.$$

Außerdem setzt man $p_{ij}^{(0)} = \delta_{ij}$.
Weiter ist

$$f_{ij}^{(s)} = Prob(X_{m+s} = j \mid X_m = i,\ X_{m+1} \neq j, \dots X_{m+s-1} \neq j)$$

die Wahrscheinlichkeit, vom Zustand i aus startend in genau s Schritten zum ersten Mal in den Zustand j zu gelangen (*first entrance probability*).
Damit ist

$$f_{ij}^{*} = \sum_{s=1}^{\infty} f_{ij}^{(s)}$$

die Wahrscheinlichkeit, jemals vom Zustand i in den Zustand j zu gelangen.
Diese Werte kann man folgendermaßen aus den Übergangswahrscheinlichkeiten berechnen: Die Wahrscheinlichkeit, in genau s Schritten von i nach j zu kommen, kann man ausdrücken als die Wahrscheinlichkeit, in σ Schritten zum ersten Mal nach j zu gelangen, und dann nach $s - \sigma$ Schritten wieder in j zu landen:

$$p_{ij}^{(s)} = \sum_{\sigma=1}^{s} f_{ij}^{(\sigma)} p_{jj}^{(s-\sigma)}.$$

Für $s = 1, 2, 3, \dots$ erhält man sukzessive die $f_{ij}^{(s)}$.

Wenn $f_{ij}^{*} = 1$ ist, so definiert $\{f_{ij}^{(s)}\}_{s \geq 1}$ eine Wahrscheinlichkeitsverteilung, die Erst-Eintritts-Verteilung (*first entrance time distribution*). Ihr Erwartungswert ist die

$$m_{ij} = \sum_{s=1}^{\infty} s f_{ij}^{(s)}.$$

Dieser Wert ist die erwartete Anzahl von Schritten bis zum ersten Eintritt in den Zustand j, wenn die Kette in i startet (*mean first passage time, mittlere Erst-Eintritts-Zeit*). Mit diesen Begriffen kann man Zustände klassifizieren. Der Zustand $i \in S$ heißt

- *periodisch* mit Periode $d > 1$, wenn $p_{ii}^{(s)} \neq 0$ nur für $s = td$, und d ist die größte natürliche Zahl mit dieser Eigenschaft.
- *aperiodisch*, wenn kein solches d existiert.
- *persistent*, wenn die Rückkehr von i nach i sicher ist, d.h. $f_{ii}^{*} = 1$.
- *transient*, wenn $f_{ii}^{*} < 1$.
- *Null-Zustand*, wenn die erwartete Zeit bis zur Wiederkehr unendlich ist: $m_{ii} = \infty$.

- *ergodisch*, wenn i aperiodisch und kein Null-Zustand ist.
- *absorbierend*, wenn i nicht verlassen werden kann, d.h. $p_{ii} = 1$.

Weiterhin kann man *Tabu-Zustände* einführen, die während einer Folge von Übergängen nicht betreten werden dürfen. Sei $H \subseteq S$ eine Tabu-Menge. Dann definiert man

$$_H p_{ij}^{(s)} = Prob(X_{m+s} = j \mid X_m = i, X_\mu \notin H, m < \mu < m+s),$$

die Wahrscheinlichkeit, in s Schritten von i nach j zu kommen, ohne zwischendrin einen Zustand aus H zu benutzen, und

$$_H f_{ij}^{(s)} = Prob(X_{m+s} = j \mid X_m = i, X_\mu \notin H, X_\mu \neq j, m < \mu < m+s)$$

die Wahrscheinlichkeit, in s Schritten erstmals j zu erreichen, wenn man bei i startet und zwischendrin keinen Zustand aus H betritt. Es gilt also

$$_H f_{ij}^{(s)} = {}_{j,H} p_{ij}^{(s)},$$

wenn man abkürzend j, H für $\{j\} \cup H$ schreibt.
Wenn $k \notin H$ ist, dann kann man in s Schritten von i nach j gelangen, indem man entweder den Zustand k gar nicht benutzt, oder ihn zum ersten Mal im σ-ten Schritt betritt, und dann noch in $s - \sigma$ Schritten von k nach j geht. Dies geht auch, wenn eine Tabu-Menge H gegeben ist:

$$_H p_{ij}^{(s)} = {}_{k,H} p_{ij}^{(s)} + \sum_{\sigma=1}^{s} {}_{k,H} p_{ik}^{(\sigma)} \; {}_H p_{kj}^{(s-\sigma)}.$$

A.2 Ergodische Markov-Ketten

Eine Markov-Kette heißt ergodisch, wenn alle Zustände ergodisch sind, also wenn jeder Zustand aperiodisch ist und die Wiederkehr nach endlich vielen Schritten sicher ist.
Ein notwendiges und hinreichendes Kriterium dafür ist, daß ein Exponent m existiert, so daß P^m nur Einträge > 0 enthält.
Eine Wahrscheinlichkeitsverteilung $\vec{q} = (q_j)_{j \in S}$ heißt stationär für die Markov-Kette P, wenn gilt

$$q_j = \sum_{i \in S} q_i p_{ij} \text{ für alle } j \in S$$

Eine ergodische Markov-Kette besitzt eine eindeutige stationäre Wahrscheinlichkeitsverteilung $\vec{q} = (q_j)_{j \in S}$ mit $q_j > 0$ für alle j. In diesem Fall gilt, daß die iterierten

Übergangsmatrizen gegen eine Grenzmatrix A konvergieren, deren Zeilen gerade die stationäre Verteilung $\vec{q}$ sind:

$$P^m \longrightarrow A = \begin{pmatrix} (q_1, q_2, \ldots, q_n) \\ (q_1, q_2, \ldots, q_n) \\ \cdots \\ (q_1, q_2, \ldots, q_n) \end{pmatrix} \quad \text{für } m \to \infty$$

Wenn die ergodische Kette P die stationäre Wahrscheinlichkeitsverteilung $(q_1, \ldots, q_n)$ besitzt, dann kann man auch die rückwärtslaufende Kette $\hat{P}$ (*time-reversed chain*) betrachten. Die Wahrscheinlichkeit, vom Zustand i einen Schritt rückwärts in den Zustand j zu machen, ist gleich der Wahrscheinlichkeit, sich in j zu befinden und von dort einen Schritt vorwärts nach i zu machen unter der Bedingung, sich jetzt im Zustand i zu befinden:

$$\hat{p}_{ij} = \frac{q_j p_{ji}}{q_i}.$$

Insbesondere ist auch $\hat{P}$ ergodisch.

Eine sinnvolle Größe für die Behandlung ergodischer Markov-Ketten ist die Fundamentalmatrix Z. Sie ist definiert als

$$\begin{aligned} Z &= (I - P + A)^{-1} \\ &= I + \sum_{n=1}^{\infty} (P - A)^n \\ &= I + \sum_{n=1}^{\infty} (P^n - A) \end{aligned} \tag{A.1}$$

Die zweite Gleichheit gilt wegen $A^2 = A$, $PA = AP = A$ und deshalb $(P - A)^n = P^n - A$.

Die erwarteten Erst-Eintritts-Zeiten m_{ij} erfüllen folgende Bedingung, die sich ergibt, wenn man die bedingten Erwartungswerte nach einem Schritt betrachtet:

$$\begin{aligned} m_{ij} &= p_{ij} + \sum_{k \neq j} p_{ik}(m_{kj} + 1) \\ &= 1 + \sum_{k \neq j} p_{ik} m_{kj} \\ &= 1 - p_{ij} m_{jj} + \sum_{k} p_{ik} m_{kj}. \end{aligned} \tag{A.2}$$

Daraus folgt

$$m_{ii} = 1/q_i, \tag{A.3}$$

indem man (A.2) mit q_i multipliziert und aufsummiert

$$\begin{aligned}\sum_i q_i m_{ij} &= 1 - \sum_i q_i p_{ij} m_{jj} + \sum_k \sum_i q_i p_{ik} m_{kj} \\ &= 1 - q_j m_{jj} + \sum_k q_k m_{kj}\end{aligned}$$

Also ist

$$q_j m_{jj} = 1$$

Damit kann man zeigen, daß die Lösung des Systems (A.2) eindeutig ist. Denn sei m'_{ij} eine zweite Lösung. Die Hauptdiagonalelemente sind gleich: $m_{ii} = m'_{ii} = 1/q_i$. Für die Differenz gilt für jede Zeile i:

$$m_{ij} - m'_{ij} = \sum_k p_{ik}(m_{kj} - m'_{kj})$$

Also muß $(m_{ij} - m'_{ij})_{j=1,\dots,n}$ eine stationäre Verteilung für P sein. Also sind alle Zeilen gleich. Da aber die Hauptdiagonalen gleich sind, muß jedes Element gleich sein: $m_{ij} - m'_{ij} = 0$.

Weiter gilt

$$m_{ij} = (\delta_{ij} - z_{ij} + z_{jj})/q_j. \tag{A.4}$$

Man rechnet nach, daß (A.4) das System (A.2) erfüllt. Wegen der Eindeutigkeit folgt die Gleichheit.

Wenn die Zugriffe unabhängig sind ($P = A$), dann gilt

$$Z = I \quad \text{und} \quad m_{ij} = 1/q_j \tag{A.5}$$

Für rückwärtslaufende Ketten gilt für $i \neq j$:

$$\hat{m}_{ij} + \hat{m}_{ji} = m_{ij} + m_{ji} \tag{A.6}$$

Denn aus der Definition von Z folgt $\hat{z}_{ij} = z_{ji} q_j / q_i$ und daraus

$$\hat{m}_{ij} - m_{ij} = (z_{ij} - \hat{z}_{ij})/q_j$$

und daraus die Behauptung.

A.3 Absorbierende Markov-Ketten

Ein Zustand i heißt *absorbierend*, wenn er nicht mehr verlassen werden kann: $p_{ii} = 1$. Wenn man eine Menge H von absorbierenden Zuständen definiert, interessiert man sich für das Verhalten der Markov-Kette in der Rest-Menge T der transienten Zustände. Sei

$$Q = P|_{T \times T}$$

die Übergangsmatrix der transienten Zustände. Eine wichtige Rolle spielt die Fundamentalmatrix der absorbierenden Kette,

$$\begin{aligned} N &= (I - Q)^{-1} \\ &= I + \sum_{n=1}^{\infty} Q^n \end{aligned}$$

Man kann fragen, wie groß die erwartete Anzahl von Schritten ist, bevor Absorption eintritt. Wenn man in $i \in T$ startet, ist die mittlere Zahl von Besuchen in $j \in T$ gegeben durch n_{ij} (siehe Lemma 1.15). Die erwartete Anzahl von Schritten, bis Absorption eintritt, ist dann $\sum_{j \in T} n_{ij}$, wenn man in i startet.

Weitere Aussagen sind in Kemeny/Snell [22] angegeben.

Literaturverzeichnis

[1] R. Ahlswede und I. Wegener. *Suchprobleme*. Teubner Verlag, Stuttgart, 1979.

[2] B. Allen und I. Munro. Self-organizing binary search trees. *Journal of the ACM*, 25(4):526–535, 1978.

[3] E. J. Anderson, P. Nash, und R. R. Weber. A counterexample to a conjecture on optimal list ordering. *Journal of Applied Probability*, 19:730–732, 1982.

[4] J. P. Arnaud. *Sur quelques problèmes concernant les librairies*. Dissertation, Université Paul Sabatier, Toulouse, 1977.

[5] J. Bell und G. Gupta. An evaluation of self-adjusting binary search tree techniques. *Software - Practice and Experience*, 23(4):369–382, 1993.

[6] M. E. Bellow. *Performance of Self-Organizing Sequential Search Heusristics under Stochastic Reference Models*. Dissertation, Carnegie-Mellon University, Pittsburgh, 1983.

[7] J. R. Bitner. Heuristics that dynamically organize data structures. *SIAM Journal of Computing*, 8(1):82–110, 1979.

[8] P. J. Burville und J. F. C. Kingman. On a model for storage and search. *Journal of Applied Probability*, 10:697–701, 1973.

[9] Ph. Chassaing. Optimality of move-to-front for self-organizing data structures with locality of references. Prépublication de l'Institut Élie Cartan, Nancy, 1993.

[10] Ph. Chassaing. Private Korrespondenz.

[11] H. T. Ch'ng, B. Srinivasan, und B. C. Ooi. Study of selforganizing heuristics for skewed access patterns. *Information Processing Letters*, 30:237–244, 1989.

[12] F. R. K. Chung, D. J. Hajela, und P. D. Seymour. Self-organizing sequential search and Hilbert's inequalities. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, Seiten 217–223, New York, 1985.

[13] K. L. Chung. *Markov Chains with Stationary Transition Probabilities*. Springer-Verlag, Berlin, zweite Auflage, 1967.

[14] W. Feller. *An Introduction to Probability Theory and its Applications*, Band 1. John Wiley & Sons, New York, dritte Auflage, 1968.

[15] G. N. Frederickson. Self-organizing heuristics for implicit data stuctures. *SIAM Journal of Computing*, 13(2):277–291, 1984.

[16] G. H. Gonnet, J. I. Munro, und H. Suwanda. Exegesis of self-organizing linear search. *SIAM Journal of Computing*, 10(3):613–637, 1981.

[17] W. J. Hendricks. The stationary distribution of an interesting markov chain. *Journal of Applied Probability*, 9:231–233, 1972.

[18] W. J. Hendricks. An extension of a theorem concerning an interesting markov chain. *Journal of Applied Probability*, 10:886–890, 1973.

[19] M. Hofri und H. Shachnai. On the optimality of the counter scheme for dynamic linear lists. *Information Processing Letters*, 37:175–179, 1991.

[20] G. Hotz. Search tress and search graphs for markov sources. *Journal of Information Processing and Cybernetics*, 29:283–292, 1993.

[21] Y. C. Kan und S. M. Ross. Optimal list order under partial memory constraints. *Journal of Applied Probability*, 17:1004–1015, 1980.

[22] J. G. Kemeny und J. L. Snell. *Finite Markov Chains*. Van Nostrand, Princeton, 1960.

[23] J. H. B. Kemperman. *The First Passage Problem for a Stationary Markov Chain*. University Press, Chicago, 1961.

[24] L. K. Konneker und Y. L. Varol. A note on heuristics for dynamic organization of data structures. *Information Processing Letters*, 12(5):213–216, 1981.

[25] K. Lam. Comparison of self-organizing linear search. *Journal of Applied Probability*, 21:763–776, 1984.

[26] K. Lam, M. Y. Leung, und M. K. Siu. Self-organizing files with dependent accesses. *Journal of Applied Probability*, 21:343–359, 1984.

[27] K. Lam, M. K. Siu, und C. T. Yu. A generalized counter scheme. *Theoretical Computer Science*, 16:271–278, 1981.

[28] J. McCabe. On serial files with relocatable records. *Operations Research*, 12:609–618, 1965.

[29] K. Mehlhorn. *Sortieren und Suchen*, Band 1 von *Datenstrukturen und effiziente Algorithmen*. Teubner Verlag, Stuttgart, zweite Auflage, 1988.

[30] P. R. Nelson. The transposition replacement policy with a partial memory constraint. *Journal of Applied Probability*, 19:733–736, 1982.

[31] R. M. Phatarfod. On the matrix occuring in a linear search problem. *Journal of Applied Probability*, 28:335–346, 1991.

[32] R. M. Phatarfod. On the transition probabilities of the move-to-front scheme. *Journal of Applied Probability*, 31:570–574, 1994.

[33] R. I. Phelps und L. C. Thomas. On optimal performance in self-organizing paging algorithms. *Journal Inf. Optimization Science*, 1:80–93, 1980.

[34] D. Richards. Studies of self-organizing placement algorithms for linear and finger lists. *Information Sciences*, 63:153–172, 1992.

[35] R. Rivest. On self-organizing sequential search heuristics. *Communications of the ACM*, 19(2):63–67, 1976.

[36] A. M. Tenenbaum und R. M. Nemes. Two spectra of self-organizing sequential search algorithms. *SIAM Journal of Computing*, 11:557–566, 1982.

Index