Technische Universität Wien

Seminararbeit

What's in YOUR wallet?

Autor:
Sophie Hermann

Supervisor:
Dr. Stefan Gerhold

28. Februar 2017

Inhaltsverzeichnis

1 Einleitung

Was ist mit "What's in your wallet?" gemeint?
Es geht um das Münzfach der Geldbörse, konkret also um die Frage, wie man einen bestimmten Betrag mit den zur Verfügung stehenden Münzen begleichen kann. [2]

Hier ist es sinnvoll zwei Perspektiven zu unterscheiden: Den Kassierer und den Kunden. Während der Kunde verschiedene Ziele bei der Wahl seiner Münzen verfolgen kann, wird der Kassierer nur eines verfolgen: Möglichst wenig Münzen pro Transaktion zu verwenden, damit er möglichst lange jeden Betrag zum Wechseln zur Verfügung hat.

2 Perspektive des Kassierers

2.1 Der Greedy-Algorithmus [1],[2]

Ziel eines Kassierers ist es, die Minimalanzahl an benötigten Münzen herauszufinden. Intuitiv verwendet er einen sogenannten "Greedy-Algorithmus":

Sei also n der Betrag und $A = (1, a_1, ..., a_n)$ die verwendete Währung.

- Er nimmt die gößte Münze $a_i \leq n$.
- Ist $a_i = n$, so ist er fertig.
- Ist $a_i < n$, so beginnt der Algorithmus von vorne mit der Differenz $n - a_i$.

Leider führt dieser simple Algorithmus nicht bei jeder Währung tatsächlich zur Minimalanzahl. Es stellt sich heraus, dass eine Währung (abhängig von der Anzahl an unterschiedlichen Münzen) einige Voraussetzungen erfüllen muss, damit das der Fall ist.

2.2 Wann ist der Greedy-Algorithmus anwendbar? [1]

Definition 2.1.
$opt_A(c)$ definieren wir als die minimale Anzahl an Münzen, die für einen Betrag c mit der Währung A benötigt wird.
$grd_A(c)$ sei wiederum die Anzahl, die der Greedy-Algorithmus benötigt.
Wir bezeichnen eine Währung A als *orderly*, wenn $\forall c > 0$ gilt: $opt_A(c) = grd_A(c)$.
Einen Betrag c, für den $opt_A(c) \neq grd_A(c)$, bezeichnen wir als *Gegenbeispiel*.

Die wahrscheinlich wichtigste Aussage liefert uns der folgende Satz:

Satz 2.2. (One-Point-Theorem)
Sei $A' = (1, a_1, ..., a_k)$ *orderly*, $a_{k+1} > a_k$ und $m := \lceil \frac{a_{k+1}}{a_k} \rceil$. Dann ist $A = (1, a_1, ..., a_k, a_{k+1})$ genau dann *orderly*, wenn $opt_A(ma_k) = grd_A(ma_k)$.

Bemerkung. Wenn wir also eine Währung A' haben, von der wir schon wissen, dass sie *orderly* ist, und zu dieser eine einzige größere Münze hinzufügen, so müssen wir für die neue Währung A nur einen einzigen Wert ma_k betrachten, um herauszufinden, ob A wieder *orderly* ist. Daher leitet sich auch der Name '*One-Point-Theorem*' für diesen Satz ab.

Beweis. Die eine Richtung ergibt sich direkt aus der Definition von *orderly* und ist somit trivial. Sei also $opt_A(ma_k) = grd_A(ma_k)$. Es gilt

$$(m-1)a_k + 1 \leq a_{k+1} \leq ma_k.$$

$\forall c < a_{k+1}$ stimmen $opt_A(c)$, $grd_A(c)$, $opt_{A'}(c)$ und $grd_{A'}(c)$ überein, also gilt insbesondere $opt_A(c) = grd_A(c)$. Alle $c \geq a_{k+1}$ teilen wir nun in zwei Gruppen: $c \in [a_{k+1}, ma_k)$ und $c \geq ma_k$.

1. $c \in [a_{k+1}, ma_k)$: Für jedes solche c gilt $c < 2a_{k+1}$ und damit enthält jede Bezahlung dieses Betrags entweder keine oder eine Münze mit dem Wert a_{k+1}. Zusammen damit, dass A' *orderly* ist ergibt das

$$opt_A(c) = \min\{1 + grd_{A'}(c - a_{k+1}), grd_{A'}(c)\}.$$

Zugleich gilt $1 + grd_{A'}(c - a_{k+1}) = grd_{A'}(c)$, also reicht es zu zeigen, dass

$$grd_A(c) \leq grd_{A'}(c),$$

um zu beweisen, dass $opt_A(c) = grd_a(c)$. Man sieht, dass

$$grd_{A'}(c) = (m-a) + grd_{A'}(c - (m-1)a_k).$$

Für die Funktion $opt_{A'} = grd_{A'}$ gilt die Dreiecks-Ungleichung, also

$$grd_{A'}(ma_k - a_{k+1}) + grd_{A'}(c - (m-a)a_k) \geq grd_{A'}(c - a_{k+1} + a_k) = 1 + grd_{A'}(c - a_{k+1}).$$

Letztendlich gilt

$$\begin{aligned} grd_{A'}(c) - grd_A(c) &= (m-1) + grd_{A'}(c - (m-1)a_k) - (1 + grd_{A'}(c - a_{k+1})) \\ &\geq m - 2 + 1 - grd_{A'}(ma_k - a_{k+1}) = m - 1 - grd_{A'}(ma_k - a_{k+1}). \end{aligned}$$

Allerdings ist

$$1 + grd_{A'}(ma_k - a_{k+1}) = grd_A(ma_k) = opt_A(ma_k) \leq m,$$

was die gewünschte Ungleichung

$$grd_{A'}(c) - grd_A(c) \geq 0$$

impliziert.

2. $c \geq ma_k$. Wir definieren uns $\mathscr{OPT}(c)$ als die Menge aller optimaler Bezahlungen für c:

$$\mathscr{OPT}(c) = \left\{ (x_0, ..., x_{k+1}) : \sum_{i=0}^{k+1} x_i a_i = c \text{ und } \sum_{i=0}^{k+1} x_i \text{ ist minimal} \right\}.$$

Es reicht Beträge $(x_i) \in \mathscr{OPT}(c)$ mit $x_{k+1} > 0$ zu betrachten. Gehen wir also von einer optimalen Bezahlung (x_i) aus. Wir haben nun zwei Möglichkeiten:

- wenn $x_k \geq m$ ist, ersetzen wir m Münzen des Wertes a_k mit denjenigen, die der Greedy-Algorithmus für diesen Betrag ma_k liefert. Die Gesamtanzahl der Münzen vermehrt sich dadurch nicht, da $opt_A(ma_k) = grd_A(ma_k)$, die Anzahl der Münzen des Wertes a_k verringert sich.
- wenn wiederum $\sum_{i=0}^{k-1} x_i a_i \geq a_k$, dann ersetzen wir diesmal die Münzen, die benötigt werden, um $\sum_{i=0}^{k-1} x_i a_i$ zu bezahlen, mit denjenigen, die man mit dem Greedy-Algorithmus (mit der Währung A' als Basis) erhält. Wieder vermehrt sich die Gesamtanzahl der Münzen nicht, da A' *orderly* ist, aber der Teilbetrag, der mit den Münzen mit den Werten $1, a_1, ..., a_{k-1}$ beglichen wird, verringert sich.

Wenn wir diese zwei Schritte oft genug anwenden, haben wir am Ende eine optimale Bezahlung (x_i), die $\sum_{i=0}^{k-1} x_i a_i < a_k$ und $x_i < m$ erfüllt. Dann gilt

$$\sum_{i=0}^{k} x_i a_i \leq a_k - 1 + (m-1)a_k = ma_k - 1 < c$$

und damit $x_{k+1} > 0$. □

Lemma 2.3. Wenn man zu einer Währung $A = (1, a_1, ..., a_k)$, die *orderly* ist, eine Münze $a_{k+1} = n \cdot a_k$, $n \in \mathbb{N}$ (also ein Vielfaches der größten Münze a_k von A) hinzufügt, so ist die neue Währung wieder *orderly*.

Beweis. Folgt direkt aus dem *One-Point-Theorem.*

□

Offensichtlich ist jede Währung $A = (1)$, die nur aus einer einzigen Münze besteht, *orderly*. Unsere kleinste Münze ist immer die 1-Cent-Münze, damit mit unseren Währungen tatsächlich jeder Betrag $c > 0$ darstellbar ist.

2.2.1 Währungen mit bis zu 5 Münzen

Proposition 2.4. Jede Währung $A = (1, a_1)$ aus zwei Münzen ist *orderly.*

Beweis. Zum Beweis verwenden wir das *One-Point-Theorem*: Wir wissen, dass $A' = (1)$ *orderly* ist und fügen nun die Münze $a_{k+1} = a_1 > 1$ hinzu. Das ergibt $m = \lceil \frac{a_1}{1} \rceil = a_1$ und damit $ma_k = a_1 \cdot 1 = a_1$. Offensichtlich gilt

$$opt_A(a_1) = grd_A(a_1) = 1,$$

da $a_1 \in A$ und damit ist jede Währung $A = (1, a_1)$ mit $a_1 > 1$ *orderly.*

□

Wie schaut es nun für eine Währung $A = (1, a_1, a_2)$ mit 3 Münzen aus?

Definition 2.5. Für $a > 0$ definieren wir:

$$\begin{aligned}\mathscr{A}(a) &= \bigcup_{m=1}^{\infty}\bigcup_{l=0}^{m}\{ma - l\} \\ &= \{a-1, a\} \cup \{2a-2, 2a-1, 2a\} \cup ... \cup \{ma-m, ..., ma\} \cup ...\end{aligned}$$

Satz 2.6. Eine Währung $A = (1, a_1, a_2)$ ist genau dann *orderly*, wenn $a_2 - a_1 \in \mathscr{A}(a_1)$.

Beweis. Sei $m = \lceil \frac{a_2}{a_1} \rceil$. Laut dem *One-Point-Theorem* ist A genau dann *orderly*, wenn man mit dem Greedy-Algorithmus für ma_k die minimale Anzahl an Münzen erhält, was äquivalent ist zu

$$grd_A(ma_1) \leq m$$

beziehungsweise

$$ma_1 - a_2 \leq m - 1,$$

was zur Folge hat, dass $a_2 - a_1 = (m-1)a_1 - (ma_1 - a_2) \in \mathscr{A}(a_1)$, genauer gesagt gehört $a_2 - a_1$ zum $(m-1)$-ten Summand von $\mathscr{A}(a_1)$.

Wenn nun m der letzte Wert ist, für den $a_2 - a_1$ zum $(m-1)$-ten Summanden von $\mathscr{A}(a_1)$ gehört, dann ist $\lceil \frac{a_2}{a_1} \rceil = m$ und $a_2 - a_1 = (m-1)a_1 - l$ für ein $l \leq m-1$. Damit folgt

$$grd_A(ma_1) = 1 + (ma_1 - a_2) = 1 + l \leq m$$

und somit die Behauptung. □

Betrachten wir nun die Differenz zwischen den Werten der Münzen einer Währung, die *orderly* ist, näher:

Proposition 2.7. Sei $A = (1, a_1, ..., a_k)$ *orderly* und $a_1 \geq 3$, dann gilt

$$a_i - a_{i-1} \neq 1$$

für alle $i = 1, ..., k$.

Beweis. Nehmen wir an, dass $a_j - a_{j-1} = 1$ und sei j der letzte Index für den das gilt. Da $a_1 \geq 3$ muss $j \geq 2$ sein. Sie m der letzte Index, für den $a_m - a_{m-1} < a_{j-1}$ gilt. Dann ist $a_{m+1} - a_m \geq a_{j-1}$ und wenn nun $a_{m-1} + a_j < a_{m+1}$ führt das zu einem Widerspruch. Also muss $a_{m+1} \leq a_{m-1} + a_j$ sein. Da $a_{m+1} - a_m \geq a_{j-1}$ folgt

$$a_j = a_{j-1} + 1 \leq (a_{m+1} - am) + (a_m - a_{m-1}) \leq a_j$$

und damit $a_m - a_{m-1} = 1$ und $a_{m+1} - a_m = a_{j-1} - 1$.
Es folgt

$$a_m < a_{m-1} + a_{j-1} < a_{m+1},$$

was bedeutet, dass eine Münze den Wert $a_{m-1} + a_{j-1} - a_m = a_{j-1} - 1$ haben muss, woraus ein Widerspruch zur Minimalität von j folgt. □

Diese Proposition kann folgendermaßen verschärft werden:

Proposition 2.8. Sei $A = (1, a_1, ..., a_k)$ *orderly*. Dann gilt

$$a_i - a_{i-1} \geq a_1 - 1$$

für alle $i = 1, ..., k$.

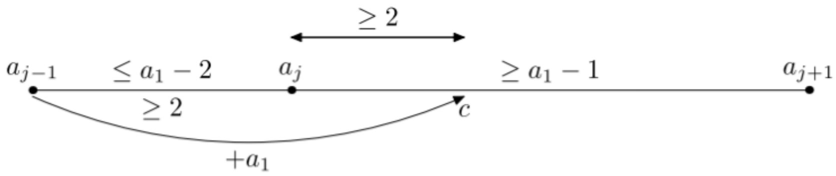

Abbildung 1: Illustration des Beweises von Proposition 2.8

Beweis. Offensichtlich gilt die Behauptung für $a_1 = 2$, also betrachten wir $a_1 \geq 3$. Sei j der größte Index, für den $a_j - a_{j-1} \leq a_1 - 2$. Wegen Proposition 2.7 wissen wir, dass $a_j - a_{j-1} \geq 2$. Wegen der Wahl von j als den größten Index haben wir $a_{j+1} - a_j \geq a_1 - 1$ (es ist möglich, dass $a_{j+1} = \infty$).
Betrachten wir nun den Betrag

$$c = a_{j-1} + a_1.$$

Es gilt $a_j + 2 \leq c \leq a_j + a_1 - 2 < a_{j+1}$, also gilt $opt_A(c) = 2$. Der Greedy-Algorithmus zerlegt den Betrag c in a_j und $c - a_j$ Münzen des Wertes 1, was zusammen eine Anzahl von

$$1 + (c - a_j) \geq 1 + 2 = 3$$

Münzen ergibt. Das ist aber ein Widerspruch dazu, dass A *orderly* ist.

□

Für den Beweis des folgenden Satzes benötigen wir noch einige Eigenschaften der Menge $\mathscr{A}(a)$:

Fakta 2.9. Sei $a \geq 2$ eine ganze Zahl. Dann gilt:

(1) wenn $x, y \in \mathscr{A}(a)$, dann ist $x + y \in \mathscr{A}(a)$.

(2) eine ganze Zahl $x \geq 2$ ist genau dann kein Element von $\mathscr{A}(a)$, wenn es eine ganze Zahl $p \geq 0$ gibt, sodass

$$pa + 1 \leq x \leq (p + 1)a - (p + 2).$$

(3) Wenn $p_1 < p_2 ... < p_m$ und $p_m - p_1 \notin \mathscr{A}(a)$, dann ist für $2 \leq j \leq m$ $p_j - p_{j-1} \notin \mathscr{A}(a)$ (folgt aus (1)).

(4) Wenn $pa < x$ und $x = (p + 1)a - c$ für ein c (kann auch negativ sein), dann impliziert $x \in \mathscr{A}(a)$: $c \leq p + 1$.

Nun können wir folgenden Satz formulieren:

Satz 2.10. Wenn $A = (1, a_1, ..., a_k)$ *orderly* ist, dann gilt $\forall \;\; 0 \leq i < j \leq k$:

$$a_j - a_i \in \mathscr{A}(a_1)$$

Beweis. Sollte dies nicht gelten, so muss nach Punkt (3) von oben ein j existieren, sodass $a_j - a_{j-1} \notin \mathscr{A}(a)$, was äquivalent ist zu

$$pa_1 + 1 \leq a_j - a_{j-1} \leq (p+1)a_1 - (p+2)$$

für gewisse p. Von allen Paaren (p, j), für die diese Ungleichungskette gilt, wählen wir nun das lexikographisch kleinste. Wenn wir nun die linke Seite mit der Rechten vergleichen erhalten wir $a_1 \geq p+3$, also $a_1 \geq 4$ und $1 \leq p \leq a_1 - 3$.
Es gilt $opt_A(a_{j-1} + (p+1)a_1) \leq p+2$ und

$$a_j + (p+2) \leq a_{j-1} + (p+1)a_1 \leq a_j + a_1 - 1,$$

demzufolge ist $grd_{\{1,..,a_j\}}(a_{j-1}+(p+1)a_1) \geq p+3$. Es folgt, dass $a_{j+1} \leq a_{j-1}+(p+1)a_1$. Dann gilt

$$a_{j+1} - a_j \leq a_{j-1} + (p+1)a_1 - a_j \leq a_1 - 1.$$

Nach Proposition 2.7 muss allerdings Gleichheit herrschen, mit anderen Worten:

$$a_{j+1} = a_{j-1} + (p+1)a_1,$$
$$a_j = a_{j-1} + pa_1 + 1.$$

Wählen wir nun das größte l, für das $a_{l+1} - a_l \leq a_{j-1} + (p-1)a_1 + 2$ gilt (solche l existieren, zum Beispiel wenn $a_{j+1} - a_j = a_1 - a$ ausreichend klein ist). Aufgrund der Wahl von l gilt $a_{l+2} - a_{l+1} \leq a_{j-1} + (p-1)a_1 + 3$ (es ist möglich, dass $a_{l+2} = \infty$). Man erhält

$$\begin{aligned} a_{l+2} - a_l &= (a_{l+2} - a_{l+1}) + (a_{l+1} - a_l) \geq a_{j-1} + (p-1)a_1 + 3 + a_1 - 1 \\ &= a_{j-1} + pa_1 + 2 \\ &= a_j + 1 \end{aligned}$$

und

$$a_{l+1} - a_l \leq a_{j-1} + (p-1)a_1 + 3 = a_j - a_1 + 2,$$

womit gilt

$$a_{l+1} + a_1 - 2 \leq a_l + a_j < a_{l+2}.$$

Das impliziert, dass $a_l + a_j = a_{l+1} + a_r$ für ein $1 \leq r < j$.

Der Rest hängt nun nur noch von den möglichen Positionen von $a_l + a_{j-1}$ ab:

Wenn $a_l + a_{j-1} > a_{l+1}$, dann gibt es nach demselben Argument einen Index $s < j-1$, für den $a_l + a_{j-1} = a_{l+1} + a_s$. In diesem Fall ist $a_r - a_s = a_j - a_{j-1} \notin \mathscr{A}(a_1)$. Nach den Eigenschaften (2) und (3) von $\mathscr{A}(a_1)$ gibt es Indizes $s < r' \leq r$ und ein p', sodass

$$p'a_1 + 1 \leq a_{r'} - a_{r'-1} \leq (p'+1)a_1 - (p'+2).$$

Die Ungleichung $a_{r'} - a_{r'-1} \leq a_j - a_{j-1}$ impliziert $p' \leq p$. Das Paar (p', r') ist lexikographisch kleiner als das Paar (p, j), was zu einem Widerspruch führt, da wir (p, j) als das kleinste Paar vorausgesetzt haben.

Wenn nun $a_l + a_{j-1} = a_{l+1}$, dann folgt

$$a_l + a_j = a_l + a_{j-1} + pa_1 + 1 = a_{l+1} + pa_1 + 1,$$

und damit $a_r = pa_1 + 1$. Dann ist $a_r - a_1 = (p-1)a_1 + 1 \notin \mathscr{A}(a_1)$, was wieder ein Widerspruch zur Minimalität vom Paar (p, j) ist.

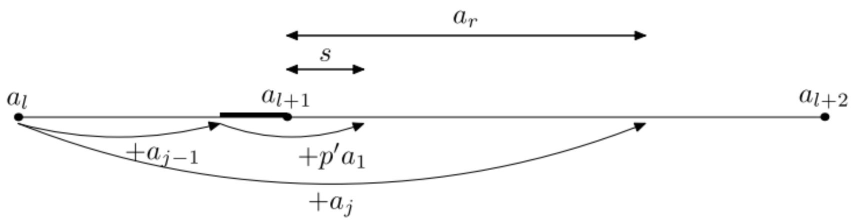

Abbildung 2: Letzter Fall: Die Länge des fett gedruckten Intervalls ist nicht in $\mathscr{A}(a_1)$

Bleibt nur noch der Fall $a_l + a_{j-1} < a_{l+1}$. a_r erfüllt

$$a_r = a_l + a_j - a_{l+1} < a_l + a_j - (a_l + a_{j-1}) = a_j - a_{j-1} = pa_1 + 1.$$

Da $a_r - a_1 < (p-1)a_1 + 1$ gilt, impliziert die Minimalität von p, dass $a_r - a_1 \in \mathscr{A}(a_1)$. Das bedeutet, dass $a_r = qa_1 - q'$ für $q \leq p$ und $0 \leq q' < q$.

Nun müssen wir zeigen, dass $a_{l+1} - (a_l + a_{j-1}) \notin \mathscr{A}(a_1)$. Es gilt

$$a_{l+1} - (a_l + a_{j-1}) = (a_l + a_j - a_r) - a_l - a_{j-1} = pa_! + 1 - a_R = (p-q)a_1 + (1+q'),$$

was mehr ist als $(p-q)a_1$, aber gleich wie $(p-q+1)a_1 - (a_1 - 1 - q')$ mit $a_1 - 1 - q' > (p+2) - 1 - q' = p + 1 - q' > p - q + 1$. Wegen Eigenschaft (2) ist $a_{l+1} - (a_l + a_{j-1})$ kein Element von $\mathscr{A}(a_1)$.

Betrachten wir nun das letzte p', für das $a_l + a_{j-1} + p'a_1 \geq a_{l+1}$. In diesem Fall gilt

$$a_l + a_{j-1} + p'a_1 < a_{l+1} + a_1 \leq a_{l+1} + a_r = a_l + a_j.$$

Offensichtlich ist $opt_A(a_l + a_{j-1} + p'a_1) \leq p' + 2$. Andererseits erhält man mit dem Greedy-Algorithmus $a_{l+1} + s \cdot 1$, wobei $s = a_l + a_{j-1} + p'a_1 - a_{l+1}$. Damit der Greedy-Algorithmus tatsächlich das optimale Ergebnis liefert, muss also gelten:

$$a + 1 \leq p' + 2,$$

also $s \leq p' + 1$. Allerdings haben wir bereits gezeigt, dass $p'a_1 - s = a_{l+1} - (a_l + a_{j-a}) \notin \mathscr{A}(a_1)$, also $s \geq p' + 1$, und damit muss letztendlich $a = p' + 1$ sein.

Betrachten wir nun $a_r - a_1$ in Bezug auf p, p' und a_1:

$$\begin{aligned} a_r - a_1 &= a_l + a_j - a_{l+1} - a_1 = a_l + (a_{j-1} + pa_1 + 1) - (a_l + a_{j-1} + p'a_1 - s) - a_1 \\ &= (pp' - 1)a_1 + (s + 1) = (p - p' - 1)a_1 + (p' + 2) \\ &= (p - p')a_1 - (a_1 - p' - 2). \end{aligned}$$

Da $a_r - a_1 \in \mathscr{A}(a_1)$ erhalten wir mit Eigenschaft (4)

$$a_1 - p' - 2 \leq p - p'$$

und damit den Widerspruch

$$a_1 < p + 3.$$

□

Satz 2.11. Wenn $A = (1, a_1, ..., a_k)$ *orderly* ist, dann ist für $2 \leq l \leq k$ die Währung $A' = (1, a_1, a_l)$ wieder *orderly*. Insbesondere ist $A' = (1, a_1, a_2)$ *orderly*.

Beweis. Da A *orderly* ist gilt wegen Proposition 2.10 $a_l - a_1 \in \mathscr{A}(a_1)$. Nach Satz 2.6 ist das ausreichend damit $A' = (1, a_1, a_l)$ *orderly* ist.

□

Definition 2.12. Eine Währung $A = (1, a_1, ..., a_k)$ bezeichnen wir als *total orderly*, wenn jede Teilwährung der Form $(1, a_1, ...a_l)$ für $l = 0, ..., k$ *orderly* ist.

Proposition 2.13. Eine Währung $A = (1, a_1, a_2, a_3)$ ist genau dann *orderly*, wenn sie *total orderly* ist.

Beweis. Eine Richtung ergibt sich direkt aus der Definition von *total orderly* und die andere folgt sofort aus dem $One - Point - Theorem$ und Satz 2.11.

□

Proposition 2.14. Eine Währung $A = (1, a_1, ..., a_4)$ ist genau dann *orderly* wenn

- (1) entweder $(1, a_1, ..., a_4) = (1, 2, a, a + 1, 2a)$ für $a \geq 4$, in diesem Fall ist $(1, a_1, a_2, a_3)$ nicht *orderly*.
- (2) oder A ist *total orderly*.

Beweis. (2) ist analog zu Beweis von 2.13. Bleibt nur noch zu zeigen, dass alle Währungen $(1, a_1, a_2, a_3, a_4)$, für die die Teilwährungen $(1, a_1, a_2, a_3)$ nicht *orderly* sind, die Form von (1) haben. Sei nun $m := \lceil \frac{a_3}{a_2} \rceil$. $(1, a_1, a_2)$ ist nach Satz 2.11 *orderly*. Wegen des *One-Point-Theorem* ist ma_2 ein Gegenbeispiel für $(1, a_1, a_2, a_3)$, also gilt $a_4 \leq ma_2$. Beide Werte $a_3 + a_3$ und $a_3 + a_2$ sind größer als ma_2, also sind sie auch größer als a_4. Daher müssen i, j mit $i < j \leq 2$ existieren sodass gilt:

$$a_3 + a_2 = a_4 + a_i,$$
$$a_3 + a_3 = a_4 + a_j.$$

Wenn wir diese beiden Gleichungen subtrahieren erhalten wir

$$a_3 - a_2 = a_j - a_i < a_j \leq a_2,$$

wodurch $a_3 < 2a_2$ folgt und damit $m = 2$.

Wir müssen zwei Fälle unterscheiden:

$j = 2$: Dann ist $a_3 - a_2 = a_2 - a_i$, also $2a_2 = a_3 + a_i$, was allerdings ein Widerspruch dazu ist, dass $(1, a_1, a_2, a_3)$ nicht *orderly* ist.

$j = 1$: Dann ist $i = 0$ und die Gleichungen 2.1 und 2.2 ergeben:

$$a_3 + a_2 = a_4 + 1$$
$$a_3 + a_3 = a_4 + a_1$$

Aus

$$a_4 + 1 = a_3 + a_2 > 2a_2 = ma_2 \geq a_4$$

folgt

$$a_4 + 1 = a_3 + a_2 = 2a_2 + 1.$$

Wenn wir nun $a_2 := a$ setzen erhalten wir $a_3 = a + 1$, $a_4 = 2a$ und $a_1 = 2a_3 - a_4 = 2$. Die Kontrolle, ob $(1, 2, a, a+1, 2a)$ *orderly* ist, funktioniert ähnlich wie der Beweis von Satz 2.2 und ergibt, dass für $a \geq 4$ die Teilwährung $(1, 2, a, a + 1)$ nicht *orderly* ist. □

Satz 2.15. Sei $(1, a_1, ..., a_k)$ *orderly*, $m \geq 2$ und

$$a_{m-1} > 2a_{m-2}, \quad a_m > 2a_{m-1}.$$

Dann gilt für alle $t \geq m$: $a_{t+1} - a_t \geq a_m - a_{m-1}$.

Beweis. Nehmen wir an, das Gegenteil sei der Fall, also $a_{t+1} - a_t \geq a_m - a_{m-1}$ für einige $t \geq m$ und sei t der kleinste Index mit dieser Eigenschaft. Sei s der größte Index für den

$$a_{s+1} - a_s < a_t - a_{m-2}.$$

Wegen der Wahl von s gilt $a_{s+1} - a_{s+1} \geq a_t - a_{m-2}$ (es ist möglich, dass $a_{s+2} = \infty$) und $a_{s+3} - a_{s+2} \geq a_t - a_{m-2}$ (wenn $a_{s+2} < \infty$). Nun haben wir zwei Fälle:

Fall 1: $a_s + a_{t+1} < a_{s+2}$. Mit dieser Annahme gilt

$$a_{s+1} < a_s + a_t < a_s + a_{t+1} < a_{s+2},$$

also existieren Indizes r und l, sodass

$$\begin{aligned} a_s + a_t &= a_{s+1} + a_r, \\ a_s + a_{t+1} &= a_{s+1} + a_l \end{aligned}$$

mit $r < l \leq t$. Das impliziert

$$a_l - a_{l-1} \leq a_l - a_r = a_{t+1} - a_t < a_m - a_{m-1}.$$

Da $l - 1 < t$ und t minimal gewählt wurde mit der Bedingung $t \geq m$ erhalten wir $l-1 < m$ mit der obigen Ungleichung. Da $l = m$ diese Ungleichung nicht erfüllt, haben wir $l \leq m - 1$ und $r \leq m - 2$, doch damit folgt

$$a_{s+1} - a_s = a_t - a_r \geq a_t - a_{m-2},$$

was zu einem Widerspruch zur Wahl von s führt.

Fall 2: Sei nun $a_s + a_{t+1} \geq a_{s+2}$. Wir beweisen nachstehende Folge von Ungleichungen:

(1) $a_{s+1} - a_s > a_{m-2}$

(2) $a_{s+2} - a_s > a_m$

(3) $a_{s+1} - a_s < a_m$

(4) $a_{s+1} - a_S \geq a_m - a_{m-1}$

(5) $a_{s+2} < a_{s+1} + a_t < a_{s+1} + a_{t+1} < a_{s+3}$.

(1) Wir haben immer

$$a_{s+2} - a_s > a_{s+2} a_{s+1} \geq a_t - a_{m-2} \geq a_m - a_{m-2} > a_{m-1}.$$

Hätten wir auch noch $a_{s+1} - a_s \leq a_{m-2}$, dann würde

$$a_{s+1} \leq a_s + a_{m-2} < a_s + a_{m-1} < a_{s+2}$$

folgen. Wie üblich bedeutet das, dass $a_{m-1} - a_{m-2} = a_l - a_r$ für $r < l \leq m - 2$ oder $a_{m-1} - a_{m-2} = a_l$ für $l \leq m - 2$. In beiden Fällen gilt dann $a_{m-1} - a_{m-2} \leq a_{m-2}$, was einen Widerspruch zu den Voraussetzungen des Satzes ergibt. Also muss $a_{s+1} - a_s > a_{m-2}$ gelten.

(2) folgt direkt aus (1) und der Wahl von s als Maximum:

$$a_{s+2} - a_s = (a_{s+2} - a_{s+1}) + (a_{s+1} - a_s) > a_t - a_{m-2} + a_{m-2} = a_t \geq a_m.$$

(3) Da wir angenommen haben, dass $a_{s+2} - a_s \leq a_{t+1}$, erhalten wir mit den Eigenschaften von s und t:

$$a_{s+1} - a_s = (a_{s+2} - a_s) - (a_{s+2} - a_{s+1}) \leq a_{t+1} - (a_t - a_{m-2}) < a_m - a_{m-1} + a_{m-2} < a_m.$$

(4) Mit (2) und (3) haben wir $a_{s+1} < a_s + a_m < a_{s+2}$, also gilt $a_s + a_m = a_{s+1} + a_r$ für $r \leq m-1$. Letztendlich folgt $a_{s+1} - a_s = a_m - a_r \geq a_m - a_{m-1}$.

(5) Mit $a_t \geq a_m$ und $2a_{m-2} < a_{m-1}$ erhalten wir

$$a_{s+3} - a_{s+1} \geq 2(a_t - a_{m-2}) = a_t + a_t - 2a_{m-2} > a_t + a_m - a_{m-1} > a_{t+1}.$$

Mit (4) und der Annahme $a_{s+2} - a_s \leq a_{t+1}$ folgt

$$a_{s+2} - a_{s+1} = (a_{s+2} - a_s) - (a_{s+1} - a_s) \leq a_{t+1} - (a_m - a_{m-1}) < a_t.$$

Damit sind (1) bis (5) bewiesen.

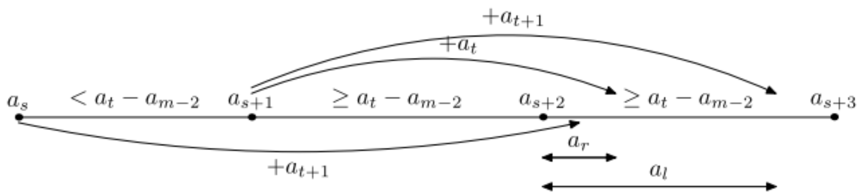

Abbildung 3: Sachverhalt von Fall 2 des Beweises von Satz 2.15

(5) impliziert die Existenz eines $r < l \leq t$ sodass

$$a_{s+1} + a_t = a_{s+2} + a_r,$$
$$a_{s+1} + a_{t+1} = a_{s+2} + a_l.$$

Wir erhalten die Ungleichung

$$a_r = a_t - (a_{s+2} - a_{s+1}) \leq a_t - (a_t - a_{m-2}) = a_{m-2}, \quad \text{also gilt } r \leq m-2,$$

was impliziert, dass

$$a_l = (a_{t+1} - a_t) + a_r < a_m - a_{m-1} + a_{m-2} < a_m, \quad \text{also } l \leq m-1.$$

Zusammen erhalten wir

$$a_{s+1} - a_s = a_{s+2} - a_s + a_l - a_{t+1} \leq a_{t+1} + a_l - a_{t+1} = a_l \leq a_{m-1}.$$

Mit (4) folgt $a_{s+1} - a_s \geq a_m - a_{m-1} > a_{m-1}$, was zu einem Widerspruch führt.

□

2.2.2 Währungen mit mehr als 5 Münzen

Wie man bis jetzt wahrscheinlich schon gemerkt hat, gibt es keine allgemein gültigen Aussagen, wann eine Währung *orderly* ist. Je mehr Münzen eine Währung hat, desto schwieriger und komplizierter wird es, eine Aussage zu erhalten, da das *One-Point-Theorem* nur anwendbar ist, solange die Währung $A = (1, a_1, ..., a_k)$ *total orderly* ist. Sobald eine Teilwährung $A' = (1, a_1, ..., a_l)$ allerdings nicht mehr *orderly* ist, sind die Voraussetzungen nicht mehr erfüllt und man kann mit diesem Satz keine Aussage mehr treffen.

Definition 2.16. Jeder Währung $A = (1, a_1, ..., a_k)$ ordnen wir nun ein Muster von $k+1$ Vorzeichen $+$ und $-$ zu, und zwar nach folgenden Kriterien:

- das i-te Vorzeichen ist ein $+$, wenn die i-te Teilwährung $A = (1, a_1, ..., a_i)$ *orderly* ist,
- umgekehrt ist das i-te Vorzeichen ein -, wenn die i-te Teilwährung $A = (1, a_1, ..., a_i)$ nicht *orderly* ist.

Das resultierende Muster bezeichnen wir als $+/-$ -Klasse.

Beispiel 2.1. Die Klasse $++$ besteht aus jeder Währung $A = \{1, a_1\}$. (siehe Proposition 2.4).

Beispiel 2.2. Jeder Währung in der Klasse $+++-+$ ist von der Form $A = \{1, 2, a, a+1, 2a\}$, wie wir in Proposition 2.14 gesehen haben.

Proposition 2.17. Jede Klasse, die mit einem $+$ endet, muss mit $+++$ beginnen.

Beweis. Diese Aussage folgt direkt aus Satz 2.11.

□

Einige Klassen von dieser Gestalt sind allerdings leer:

Proposition 2.18. Die Klasse $+++-+-+$ ist leer.

Beweis. Sei $A = (1, a_1, a_2, a_3, a_4, a_5, a_6)$ eine Währung, die zur Klasse $+++-+-+$ gehört. Wegen Proposition 2.14 (1) muss A folgende Form haben:

$$A = (1, 2, a, a+1, 2a, a_5, a_6)$$

für $4 \leq a$, $2a < a_5 < a_6$. Nach dem *One-Point-Theorem* ist ein Vielfaches von $2a$ ein Gegenbeispiel für $\{1, 2, a, a+1, 2a, a_5\}$. Diese Währung mit a_6 zu erweitern löst dieses Problem, da wir wissen dass A *orderly* ist, also muss gelten:

$$a_6 - a_5 < 2a.$$

Außerdem existieren r, s, sodass

$$a_5 + a_5 = a_6 + a_r,$$
$$a_5 + 2a = a_6 + a_s$$

mit $a_r \leq 2a$, $a_s \leq a+1$ und $1 \leq a_s < a_r$. Durch Subtrahieren der beiden Gleichungen erhalten wir $a_5 - 2a = a_r - a_s$. Die möglichen Differenzen $a_r - a_s$ $(0 \leq s < r \leq 4)$ bilden die Menge

$$\{1, a-2, a-1, a, 2a-a, 2a-1, 2a\},$$

was folgende mögliche Werte für a_5 ergibt: $2a+1$, $3a-2$, $3a-1$, $3a$, $4a-2$, $4a-1$ und $4a$. Die Werte $3a-1$, $3a$, $4a-2$, $4a-1$ und $4a$ können wir allerdings ausschließen, da damit $(1, 2, a, a+1, 2a, a_5)$ *orderly* wäre, was mit dem *One-Point-Theorem* leicht nachzuweisen ist. Bleibt für a_5 also nur noch $2a+1$ und $3a-2$.

Für $a_5 = 2a + 1$ scheitert der Greedy-Algorithmus (mit Basis $(1, ..., a_5)$) schon für $3a = 2a + a$, demzufolge gilt $a_6 \leq 3a$. Andererseits können alle drei Beträge $2a + 2a$, $2a + (2a + 1)$ und $(2a + 1) + (2a + 1)$ mit nur zwei Münzen beglichen werden und damit müssen $4a - a_6$, $4a + 1 - a_6$ und $4a + 2 - a_6$ drei aufeinander folgende Werte sein, die alle kleiner als a_6 sind. Das ist aber nur möglich, wenn $a_6 = 4a$, was zu einem Widerspruch führt.

Sei nun $a_5 = 3a - 2$. Damit $2a + (a + 1) = 3a + 1$ kein Gegenbeispiel ist muss $3a - 1 \leq a_6 \leq 3a + 1$ gelten.
Wenn $a_6 = 3a - 1$, dann ist

$$4a - 2 = (3a - 2) + a = (3a - 1) + (a - 1)$$

ein Gegenbeispiel, da es keine Münze mit dem Wert $a - 1$ gibt.
Wenn $a_6 = 3a$, so ist

$$4a - 1 = 2a + 2a = (3a - 2) + (a + 1) = 3a + (a - 1)$$

aus denselben Gründen wie vorher wieder ein Gegenbeispiel.
Ist letztendlich $a_6 = 3a + 1$, dann ist

$$4a = 2a + 2a = (3a + 1) + (a - 1)$$

ein Gegenbeispiel.

$\square$

Jede Menge $P = \{i_0, i_1, ..., i_l\} \subset \{0, 1, ..., k\}$ mit $0 = i_0 < i_1 < ... < i_l$ bestimmt eine Teilwährung $\{a_{i_0}, a_{i_1}, ..., a_{i_l}\}$ einer Währung $A = (1, a_1, ..., a_k)$.
Nach Satz 2.11 wissen wir, dass für eine Währung A, die *orderly* ist, die Teilwährung $P = \{0, 1, l\}$ mit $2 \leq l \leq k$ wieder *orderly* ist. Betrachten wir das nun näher:

Definition 2.19. Eine Menge P von der oben gegebenen Form heißt *hereditery*, wenn für jede Währung $A = (1, a_1, ..., a_k)$, die *orderly* ist, die Teilwährung, die durch P bestimmt wird, wieder *orderly* ist.

Betrachten wir nun einige interessante Klassen von Teilmengen von $\{0, 1, ..., k\}$:

Typ 1: die Menge $\{0\}$, die nur aus einem einzigen Element besteht,

Typ 2: die Mengen $\{0, l\}$ für $1 \leq l \leq k$,

Typ 3: die Mengen $\{0, 1, l\}$ für $2 \leq l \leq k$,

Typ 4: die Mengen $\{0, 1, 2, l\}$ für $4 \leq l \leq k$,

Typ 5: die ganze Menge $\{0, 1, ..., k\}$.

Die Menge $\{0, 1, 2, 3\}$ ist eine interessante Ausnahme: Sie ist nicht vom Typ 4. Betrachten wir zum Beispiel die Währung $A = (1, 2, a, a + 1, 2a)$ für $a \geq 4$. Die Teilwährung, die $P = \{0, 1, 2, 3\}$ bestimmt, ist $(1, 2, a, a + 1)$, von der wir wissen, dass sie nicht *orderly* ist.

Lemma 2.20. Für $l \geq 3$ sei B_l die Währung

$$B_l = (1, 2, 3, ..., l-1, 2l-2, 2l-1, 4l-4)$$

mit $a_l = 2l - 1$. Dann ist B_l vom Typ $+++...+-+$.

Beweis. Die Währung $(1, 2, 3, ..., l-1)$ ist klarerweise *orderly*. Fügen wir nun die Münze $2l-2$ hinzu, erhalten wir nach dem *One-Point-Theorem* wieder eine Währung, die *orderly* ist. Erweitert mit $2l-1$ ergibt das eine Währung, die nicht *orderly* ist, da $2 \cdot 2(l-1) = 4l-4$ das kleinste Gegenbeispiel ist. Zu zeigen, dass die gesamte Währung B_l *orderly* ist funktioniert analog wie im Beweis vom *One-Point-Theorem*. □

Lemma 2.21. Für $m > l \geq 2$ und $p \geq 1$ sei $A_{l,m}(p)$ die Währung:

$$\begin{aligned} A_{l,m}(p) &= (a_0, a_1, a_2, ..., a_{l-1}, a_l, a_{l+1}, a_{l+2}, ..., a_m) \\ &= (1, 2, 3, ..., l, pl, (2p-1)l, (3p-2)l, ..., ((m-l+1)p - (m-l))l) \end{aligned}$$

mit $a_l = pl$. Diese Währung ist *orderly*. Außerdem gilt wenn $p > m - l$, dass $\lceil \frac{a_m}{a_l} \rceil = m - l + 1$.

Beweis. Diese Währung ist sogar vom Typ $++++...+++$, was sehr leicht durch wiederholtes Anwenden des *One-Point-Theorems* nachgewiesen werden kann:
Um nachzuprüfen, ob $(1, a_1, ..., a_{l+i})$ für $i \geq 1$ *orderly* ist, genügt es folgendes zu betrachten:

$$\begin{aligned} 2a_{l+i-1} &= 2(pi - (i-1))l \\ &= (p(i+1) - i)l + (p(i-1) - (i-2))l \\ &= a_{l+i} + a_{l+i-2}. \end{aligned}$$

Um die zweite Aussage zu beweisen, stellen wir fest, dass

$$a_m = ((m-l+1)p - (m-l))l < (m-l+1)pl = (m-l+1)a_l$$

und wenn $p > m - l$

$$a_m = ((m-l+1)p - (m-l))l = (m-l)pl + l(p - (m-l)) > (m-l)a_l.$$

□

Angenommen wir wollen zeigen, dass eine Menge $P = \{i_0, i_1, ..., i_l\} \subset \{0, 1, ..., k\}$ nicht *hereditary* ist, so können wir Lemma 2.3 verwenden:
Zuerst suchen wir eine kleinere Währung $A' = (1, a_1, ..., a_r)$, die *orderly* ist, sodass die Teilwährung, die durch $P' = \{i_0, i_1, ..., i_{r'}\} \subset \{0, 1, ..., r\}$ bestimmt ist, nicht *orderly* ist (hier ist $r' < r \leq k$) und es gilt $i_{r'+1} > r$ oder $r' = l$. Sei c ein Gegenbeispiel für diese Teilwährung und m eine Zahl, für die $ma_r > c$ gilt. Dann ist die Währung

$$A = (1, a_1, ..., a_r, ma_r, 2ma_r, ..., (k-r)ma_r)$$

orderly und ihre Teilwährung, die durch P bestimmt ist, nicht, da alle hinzugefügten Münzen zu groß sind, um das Problem mit c zu beseitigen.

Satz 2.22. Alle Mengen P, die nicht vom Typ 1, 2, 3, 4 oder 5 sind, sind nicht *hereditery*.

Beweis. Sei $P = \{i_0, ..., i_s\} \subset \{0, ..., k\}$, $i_0 = 0$ so eine Menge. Sei r der größte Index, für den $i_r = r$ gilt (z.B. $\{0, ..., r\} \subset P$, $r+1 \notin P$). Betrachten wir nun unterschiedliche Fälle:

Fall 1: $3 \leq r < k$. Hier betrachten wir die Währung B_r, von der wir wissen, dass sie *orderly* ist. Ihre Teilwährung $(1, a_1, ..., a_r)$ ist nicht *orderly*. Wenn $r = k - 1$, sind wir fertig, wenn $r < k - 1$ müssen wir B_r zu einer Währung mit $k + 1$ Münzen, die *orderly* ist, so erweitern, wie vorher beschrieben. Die resultierende Währung hat nun eine durch P definierte Teilwährung, die nicht *orderly* ist.

Fall 2: $r = 2$. In diesem Fall ist $|P| \geq 5$, da sonst P von der Form $\{0, 1, 2\}$ oder $\{0, 1, 2, l\}$ für ein $l \geq 4$ wäre und damit vom Typ 3 oder 4. Wir betrachten nun $A_{l,m}(p)$ für $l = i_3 \geq 4$, $m = i_4$ und $p > m - l$. Deren Teilwährung

$$(1, 2, 3, a_l, a_m)$$

ist nicht *orderly*, da der Greedy-Algorithmus den Betrag

$$\lceil \frac{a_m}{a_l} \rceil a_l = (m - l + 1)a_l$$

mit einer Münze a_m und einigen der Münzen 1, 2 und 3 begleicht und damit mindestens

$$1 + \frac{(m - l)l}{3} > 1 + (m - l)$$

Münzen benötigt, was mehr ist, als wenn wir mit $m - l + 1$ Münzen a_l bezahlen. Nun genügt es, diese Währung wie vorher zu einer Währung mit $k+1$ Münzen zu erweitern.

Fall 3: $r = 1$. Hier ist $|P| \geq 4$, da ansonsten P die Form $\{0, 1, l\}$ hätte, und damit zu Typ (3) gehören würde. Betrachten wir nun $A_{l,m}(p)$ für $l = i_2 \geq 3$, $m = i_3$ und $p > m - l$. Die Teilwährung $(1, 2, a_l, a_m)$ ist aus denselben Gründen wie vorher nicht *orderly*:
Der Betrag $\lceil \frac{a_m}{a_l} \rceil a_l = (m-l+1)a_l$ wird nach dem Greedy-Algorithmus mit mindestens $1 + \frac{(m-l)l}{2} > 1 + (m - l)$ Münzen beglichen.

Fall 4: $r = 0$. Klarerweise ist $|P| \geq 3$, da Mengen der Form $\{0\}$ und $\{0, l\}$ vom Typ 1 und 2 sind. Wir verwenden wieder dasselbe Argument für $A_{l,m}(p)$ mit $l = i_1 \geq 2$, $m = i_2$ und $p > m - l$ mit der Teilwährung $(1, a_l, a_m)$: Diesmal benötigt der Greedy-Algorithmus für den Betrag $\lceil \frac{a_m}{a_l} \rceil a_l = (m-l+1)a_l$ mindestens $1 + \frac{(m-l)l}{1} > 1 + (m-l)$ Münzen.

□

Wir gehen davon aus, dass Mengen vom Typ (4) tatsächlich *hereditary* sind, obwohl das noch nicht bewiesen werden konnte:

Annahme 2.23. Wenn eine Währung $A = (1, a_1, ..., a_k)$ *orderly* ist, dann ist die Währung $(1, a_1, a_2, a_l)$ für $4 \leq l \leq k$ wieder *orderly*.

Allerdings ist es gelungen diese Aussage für einige zusätzliche Bedingungen zu zeigen:

Satz 2.24. Annahme 2.23 ist korrekt, wenn zusätzlich $a_2 > 2a_1$ und $a_3 > 2a_2$ gilt.

Beweis. Wir überprüfen nun mithilfe von Proposition 2.13, ob $(1, a_1, a_2, a_l)$ *orderly* ist. Sei $m = \lceil \frac{a_l}{a_2} \rceil$. Aufgrund von Satz 2.15 erhalten wir für $l \geq 3$ die erste der folgenden Ungleichungen:

$$a_{l+1} - a_l \geq a_3 - a_2 > a_2 > ma_2 - a_l.$$

Somit erhalten wir $a_{l+1} > ma_2$, also gibt es keine Münze zwischen a_l und ma_2 und der Greedy-Algorithmus (mit Basiswährung A) benötigt für den Betrag ma_2 nur die Münzen 1, a_1 und a_l. Somit erhalten wir die erste Gleichheit:

$$grd_{(1,a_1,a_2,a_l)}(ma_2) = grd_A(ma_2) = opt_A(ma_2) \leq opt_{(1,a_1,a_2,a_l)}(ma_2)$$

und wir erhalten mit Proposition 2.13 die Aussage.

□

2.3 Beispiel US-Dollar

Betrachten wir unsere Resultate nun für eine reale Währung, nämlich den US-Dollar:

$$\$ = (1, 5, 10, 25)$$

Die Frage, die sich nun unmittelbar stellt, ist natürlich jene, ob der Dollar *orderly* ist, also ob $opt_\$(c) = grd_\$(c) \ \ \forall c > 0$.

Proposition 2.25. Der Dollar $\$ = (1, 5, 10, 25)$ ist *orderly*.

Beweis. Nach Satz 2.6 ist eine Währung $A = (1, a_1, a_2)$ *orderly*, wenn $a_2 - a_1 \in \mathscr{A}(a_1)$. Sei nun $\$' = \{1, 5, 10\}$.

$$\mathscr{A}(5) = \{4, 5\} \cup \{8, 9, 10\} \cup ...$$

$a_2 - a_1 = 10 - 5 = 5$ ist Element von $\mathscr{A}(5)$, also ist $\$'$ *orderly*.
Um von der Währung $\$'$ nun auf den Dollar schließen zu können, verwenden wir das *One-Point-Theorem*:
Wir fügen eine Münze $a_{k+1} = 25 > 10 = a_k$ zu einer Währung $\$'$, die *orderly* ist, hinzu, also sind alle Vorraussetzungen für den Satz erfüllt. Wir erhalten $m = \lceil \frac{a_{k+1}}{a_k} \rceil = \lceil \frac{10}{5} \rceil = 2$ und damit $ma_k = 2 \cdot 5 = 10$. Offensichtlich gilt

$$opt_\$(10) = grd_\$(10) = 1,$$

da $10 \in \$$ und somit ist der Dollar tatsächlich *orderly*! □

2.4 Beispiel Euro

Wir betrachten zuerst einmal nur eine Währung €′, die nur aus den Cent-Münzen besteht:

$$€' = (1, 2, 5, 10, 20, 50)$$

Da der Euro viel mehr Cent-Münzen als der Dollar hat, ist es ein bisschen aufwendiger herauszufinden, ob der Euro *orderly* ist.

Proposition 2.26. Die Währung €' ist *orderly.*

Beweis. Betrachten wir zuerst die Teilwährung $(1, 2, 5)$. Nach Proposition 2.6 ist diese Währung genau dann *orderly*, wenn $a_2 - a_1 \in \mathscr{A}(a_1)$.

$$\mathscr{A}(2) = \{1, 2\} \cup \{2, 3, 4\} \cup ...$$

Es gilt $a_2 - a_1 = 5 - 2 = 3 \in \mathscr{A}(2)$, damit ist $(1, 2, 5)$ *ordely.*
Nach Lemma 2.3 sind die nächsten beiden Teilwährungen $(1, 2, 5, 10)$ und $(1, 2, 5, 10, 20)$ wieder *orderly*, da wir jedes Mal nur ein Vielfaches der größten Münze hinzufügen.
Um nun festzustellen, ob €' *orderly* ist, müssen wir nur das *One-Point-Theorem* anwenden: $m = \lceil \frac{a_k+1}{a_k} \rceil = \lceil \frac{50}{20} \rceil = 3$ und damit $ma_k = 3 \cdot 20 = 60$. Es folgt

$$opt_{€'}(60) = 2 = grd_{€'}(60),$$

damit ist €' *orderly.*

□

Dass die gesamte Währung € = $(1, 2, 5, 10, 20, 50, 100, 200)$ *orderly* ist, folgt trivialerweise wieder aus Lemma 2.3, da wir wieder jedes Mal nur ein Vielfaches der größten Münze hinzufügen.

3 Perspektive des Kunden am Beispiel US-Dollar [2]

Wie schon am Anfang erwähnt, gibt es viele unterschiedliche Strategien, die ein Kunde bei der Wahl seiner Münzen verfolgen kann. Wir werden hier auf einige eingehen und die resultierenden Verteilungen am Ende vergleichen.

Bevor wir jedoch beginnen, müssen wir vorweg einige Annahmen treffen:

(1) Der Cent-Anteil der Preise ist zwischen 0 und 99 gleichverteilt.

(2) Die zugrunde liegende Währung ist der US-Dollar: $\$ = (1, 5, 10, 25)$

(3) Die Anzahl der Münzen, die der Kassierer zum Wechseln verwendet, ist immer minimal und wird mit dem Greedy-Algorithmus ermittelt.

(4) Kassierer haben immer genügend von jeder Münze zum Wechseln zur Verfügung.

Mit dem US-Dollar als Währung ist für jeden Betrag $c > 0$ die minimale Partition eindeutig. Kommen wir nun zu den unterschiedlichen Ausgabestrategien:

3.1 Der *Coin-Keeper*

Die wohl einfachste Variante ist gar keine Münzen zu verwenden und somit bei fast jedem Einkauf Wechselgeld zu erhalten. Mit der Zeit werden so viele Münze zusammenkommen. Was ist nun deren Verteilung?
Aufgrund unserer Annahme, dass der Cent-Anteil der Preise gleichverteilt ist, müssen wir nur die Münzen zusammenzählen, die ein *Coin-Keeper* bei jeder der 100 Möglichen Transaktionen erhält: Wir zählen 150 Quarters, 80 Dimes, 40 Nickel und 200 Pennies.

Bevor wir nun weitere Ausgabestrategien untersuchen, machen wir noch einige Beobachtungen: Der Zustand unserer Geldbörse, wenn wir einkaufen gehen, hängt nur vom

- Zustand der Geldbörse bevor wir bezahlen,
- vom Preis der Artikel in unserem Einkaufswagen und
- von unserer Münzausgabestrategie

ab. Wir haben also eine Markov-Kette in diskreter Zeit: Das System ist die Geldbörse und das zufällige Ereignis der zu bezahlende Preis.

Sei $S = \{s_1, s_2, ...\}$ die Menge der möglichen Zustände meiner Geldbörse. Eine Markov-Kette mit endlich vielen Zuständen hat eine $|S| \times |S|$ Übergangsmatrix M deren Eintrag m_{ij} die Übergangswahrscheinlichkeit des Zustandswechsels von s_i nach s_j angibt. Sei $v = (v_1, ..., v_{|S|})$ die Anfangsverteilung unserer Markov-Kette, dann erhält man mit vM^n einen Vektor, dessen i-ter Eintrag die Wahrscheinlichkeit angibt, dass das System nach n Schritten im Zustand s_i ist.

Das Langzeitverhalten unserer Geldbörse wird demnach durch vM^n für große n beschrieben. Existiert $p = \lim_{n\to\infty} vM^n$, so gibt es eine eindeutige Antwort auf die Frage 'Was ist die erwartete Anzahl an Münzen in unserer Geldbörse?': Der i-te Eintrag von

p ist die Langzeit-Wahrscheinlichkeit, dass sich das System im Zustand s_i befindet. Wenn dieser Grenzwert unabhängig von der Anfangsverteilung v ist, erhalten wir mit p sogar die Grenzverteilung für jede beliebige Geldbörse, deren Besitzer dieselbe Ausgabestrategie verfolgt.

Angenommen also so ein p existiert, wie können wir es dann berechnen?
Die Grenzverteilung ändert sich durch Multiplikation mit M nicht, da es sonst nicht die Grenzverteilung wäre, also muss $pM = p$ gelten. Damit ist p ein linker Eigenvektor von M mit Eigenwert 1. Natürlich kann es viele solche Eigenvektoren geben, wir wissen allerdings, dass zusätzlich $p_1 + ... + p_{|S|}$ gelten muss.

Unter gewissen Voraussetzungen erhalten wir mit dem Satz von Perron-Frobenius eine Aussage über die Existenz und Eindeutigkeit eines solchen p. Unsere Markov-Kette muss zwei Bedingungen erfüllen: Sie muss irreduzibel und aperiodisch sein.
Eine Markov-Kette ist irreduzibel, wenn es für je zwei Zustände s_i und s_j ein $n \in \mathbb{N}$ gibt, sodass die Wahrscheinlichkeit in n Schritten von s_i nach s_j zu gelangen größer 0 ist. Aufgrund der Annahmen (1) und (3) und der noch folgenden Details zu den jeweiligen Ausgabestrategien ist unsere Markov-Kette tatsächlich irreduzibel.
Eine Markov-Kette ist periodisch, wenn es einen Zustand s_i gibt, sodass ein Zustandswechsel von s_i nach s_i in einem Vielfachen von $k > 1$ Schritten stattfindet. In unserem System führt eine Transaktion, deren Cent-Anteil 0 ist, zu einem direkten Zustandswechsel von s_i zu s_i und damit ist unsere Markov-Kette aperiodisch.
Eine weitere Voraussetzung für den Satz von Perron-Frobenius ist $M \geq 0$, da aber die Einträge in unserer Matrix Wahrscheinlichkeiten entsprechen, ist diese Voraussetzung trivialerweise erfüllt.
Nach dem Satz von Perron-Frobenius existiert p und ist der größte linke Eigenvektor von M mit Eigenwert 1.

Mit p können wir nun einiges berechnen:
Die erwartete Anzahl der Münzen in der Geldbörse ist

$$\sum_{i=1}^{|S|} p_i |s_i|.$$

Der erwartete Gesamtbetrag der Geldbörse, in Cent, ist

$$\sum_{i=1}^{|S|} p_i \sigma(s_i),$$

wobei $\sigma(s_i)$ die Summe der Elemente in s_i angibt.

Während der Kassierer nach Annahme (4) immer genügend von jeder Münze zur Verfügung hat, ist das für einen Kunden nicht immer der Fall. Wenn nicht jede Münze zur Verfügung steht, kann es passieren, dass der Greedy-Algorithmus nicht mehr optimal funktioniert. Ist der Zustand der Geldbörse zum Beispiel $\{25, 10, 10, 10\}$ und der zu begleichende Betrag 30 Cent, kommt man mit dem Greedy-Algorithmus nicht auf $\{10, 10, 10\}$.

3.2 Der *Minimalist-Spender*

Ein *Minimalist-Spender* wählt seine Münzen so, dass sich nach der Transaktion möglichst wenig Münzen in seiner Geldbörse befinden. Wenn der Wert der Geldbörse vor dem Bezahlen also n Cent und der zu bezahlende Preis c Dollar ist, so hat der *Minimalist-Spender* nach der Transaktion eine minimale Partition von $n - c \mod 100$ Cent. Da diese Partition eindeutig ist, ist der Zustand der Geldbörse nach dem Einkauf eindeutig bestimmt. Es gibt 100 mögliche Zustände, einen für jedes $0 \leq n \leq 99$. Nach Annahme (1) sind die Übergangswahrscheinlichkeiten für jeden Zustand $1/100$, und damit ist auch auf lange Zeit jeder Zustand gleich wahrscheinlich.
Wir erhalten also die erwartete Anzahl an Münzen $\frac{1}{100}\sum_{i=1}^{100}|s_i| = 4.7$ und den erwarteten Gesamtbetrag $\frac{1}{100}\sum_{i=0}^{99} n = 49.5$ Cent. Wenn wir nun die vorkommenden Münzen jedes Wertes für jede der 100 minimalen Partitionen von $0 \leq n \leq 99$ zusammenzählen, erhalten wir 1.5, 0.8, 0.4 und 2 als erwartete Anzahl an Quarters, Dimes, Nickel und Pennies.
Diese Ausgabestrategie ist tatsächlich jene mit der kleinsten erwarteten Anzahl an Münzen in der Geldbörse: Sei $g(n)$ die Anzahl der Münzen, die der Greedy-Algorithmus für einen Betrag n benötigt. Betrachten wir eine Strategie, für die eine irreduzible, aperiodische Markov-Kette vorliegt und sei $e(n)$ die auf lange Zeit erwartete Anzahl an Münzen in der Geldbörse in Abhängigkeit vom Gesamtbetrag n. Da $g(n)$ die minimale Anzahl an benötigten Münzen liefert, gilt $e(n) \geq g(n)$ für alle $0 \leq n \leq 99$. Da der Preis c gleichverteilt ist, ist auch der Gesamtbetrag n der Geldbörse gleichverteilt und damit ist die auf lange Zeit erwartete Anzahl an Münzen

$$\frac{1}{100}\sum_{n=0}^{99} e(n) \geq \frac{1}{100}\sum_{n=0}^{99} g(n) = \frac{47}{10}.$$

3.3 Der *Big-Spender*

Die Ausgabestrategie eines *Big-Spenders* sieht folgendermaßen aus:

(5) Er bezahlt nur dann mit Münzen, wenn er genügend zur Verfügung hat, ansonsten bezahlt er mit Scheinen und erhält Rückgeld.

(6) Hat er genug Münzen zur Verfügung, so bezahlt er so genau wie möglich.

(7) Gibt es trotzdem noch mehrere Möglichkeiten, so favorisiert er $\{a_1, ...a_m\}$ über $\{b_1, ..., b_n\}$, wobei $a_1 < ... < a_m$ und $b_1 < ... < b_n$, wenn $a_1 = b_1, ..., a_i = b_i$ und $a_{i+1} > b_{i+1}$ für mindestens ein i.

Nun müssen wir noch prüfen, ob wir endlich viele Zustände haben.

Lemma 3.1. Wir gehen nun von einer Ausgabestrategie aus, die (5) und (6) erfüllt. Hat dieser Kunde vor einer Transaktion bis zu 99 Cent , so hat er nach der Transaktion wieder maximal 99 Cent.

Beweis. Sei $0 \leq c \leq 99$ der Preis und n der Betrag in der Geldbörse des Kunden.

Wenn $c \leq n$, so folgt mit Annahme (6), dass er mindestens c Cent ausgibt und eventuell Rückgeld erhält. Damit hat der Kunde nach der Transaktion $n - c$ Cent und da $n \leq 99$ und $c \geq 0$, folgt $n - c \leq 99$.

Wenn jedoch $c > n$, so werden nach Annahme (5) keine Münzen ausgegeben sondern nur Scheine, und der Kunde erhält ein Rückgeld von $100 - c$. Damit hat er $n + 100 - c = 100 - (c - n)$ Cent nach der Transaktion und da $c > n$ folgt $c - n \geq 1$ und damit $100 - (c - n) \leq 99$.

□

Hat ein *Big-Spender* am Anfang mehr als 99 Cent, so bezahlt er nach Annahme (5) solange mit Münzen, bis er weniger als 99 Cent hat. Damit ist jeder Zustand mit mehr als 99 Cent transient und hat eine Lang-Zeit-Wahrscheinlichkeit von 0. Da es endlich viele Möglichkeiten gibt, maximal 99 Cent zu besitzen, ist damit unser Zustandsraum tatsächlich endlich.
Alle möglichen Zustände sind jene mit maximal 99 Cent und somit maximal 3 Quarters, 9 Dimes, 19 Nickel und 99 Pennies. Von diesen $4 \times 10 \times 20 \times 100 = 80000$ Zuständen haben 6720 tatsächlich einen Gesamtbetrag ≤ 99. Um unsere 6720×6720 Übergangsmatrix M zu erhalten, müssen wir alle 6720×100 möglichen Transaktionen simulieren und obwohl wir wissen, dass unser p existiert, müssen wir jenes auch erst berechnen. Um Rechenzeit zu sparen verwenden wir numerische Methoden um p anzunähern, genauer gesagt bedienen wir uns der im ARPACK Paket von Mathematica vorimplementierten Arnoldi-Iteration. Die Arnoldi-Iteration ist eine effiziente Methode, um die größten Eigenwerte samt zugehörigen Eigenvektoren einer Matrix herauszufinden.
Wir erhalten als erwartete Anzahl an Münzen 10.05, zusammengesetzt aus 1.06 Quarters, 1.15 Dimes, 0.91 Nickel und 6.92 Pennies. Der erwartete Gesamtbetrag ist wieder 49.5, was am ersten Blick vielleicht seltsam erscheint, da die Ausgabestrategien eines *Minimalist-Spenders* und eines *Big-Spenders* sehr unterschiedlich sind, allerdings ist das nur eine Folge von Annahme (1), nämlich dass alle Preise gleichverteilt sind. Wenn wir alle Informationen der Geldbörse eines *Big-Spenders* bis auf den Gesamtbetrag ignorieren, erhalten wir wieder eine Markov-Kette mit 100 Zuständen, die alle gleich wahrscheinlich sind und damit einen erwarteten Gesamtbetrag, der dem Durchschnitt aller möglichen Gesamtbeträge entspricht. Genau genommen hat jede Ausgabestrategie, bei der alle möglichen Gesamtbeträge gleich wahrscheinlich sind, den erwarteten Gesamtbetrag 49.5.

3.4 Der ***Pennies-First-Big-Spender***

Während der *Minimalist-Spender* im Durchschnitt 4.7 Münzen bei sich trägt, hat ein *Big-Spender* wesentlich mehr. Diese Differenz können wir verkleinern, wenn wir uns wie ein *Pennies-First-Big-Spender* verhalten:
Ein *Pennies-First-Big-Spender* berechnet sich als erstes den Preis modulo 5. Hat er genug Pennies, um diesen Betrag zu bezahlen, so zögert er nicht das zu tun und muss damit nur mehr die Restdifferenz begleichen. Danach verhält er sich genauso wie ein *Big-Spender*. Hat ein *Pennies-First-Big-Spender* vor der Transaktion weniger als 5 Pennies, so hat er nach der Transaktion wieder weniger als 5 Pennies. Somit trägt er nie mehr als 4 Pennies bei sich und der Zustandsraum verkleinert sich zu 1065 möglichen Zuständen. Wir erhalten 5.74 als erwartete Anzahl an Münzen, was nur eine Münze mehr als beim *Minimalist-Spender* ist. Die erwartete Anzahl an Quarters, Dimes, Nickel und Pennies ist 1.12, 1.27, 1.35 und 2.00.
Tatsächlich ist die erwartete Anzahl an Pennies genau 2, da wir uns den *Pennies-First-Big-Spender* als zwei verschiedene Personen vorstellen können: Einen *Big-Spender*, der

die Quarters, Dimes und Nickel verwaltet und eine zweite Person, die nur für die Pennies zuständig ist. Zuerst bezahlt also die Person mit den Pennies entweder wenn möglich für $c \mod 5$, oder sie erhält $5 - c \mod 5$ Pennies Wechselgeld. Da dafür keine Informationen von der anderen Person nötig sind, sind alle fünf möglichen Zustände gleich wahrscheinlich und damit ist die erwartete Anzahl an Pennies $\frac{1}{5}\sum_{i=0}^{4} i = 2$.

3.5 Vergleich mit U.S. Mint 2014

Zum Abschluss vergleichen wir unsere Ergebnisse mit der U.S. Mint 2014, also mit den 2014 in den USA in Umlauf gebrachten Münzen:

	U.S. Mint in Mio.	U.S. Mint	C-K	M-S	B-S	P-F-B-S
Quarters	1580	11.9%	31.9%	31.9%	10.6%	19.51%
Dimes	2302	17.4%	17%	17%	11.5%	22.13%
Nickels	1206	9.1%	8.5%	8.5%	9.1%	23.52%
Pennies	8146	61.6%	42.6%	42.6%	68.9%	34.84%

Wie man sieht kommen die Werte des *Big-Spenders* jenen der U.S. Mint am nächsten. Auffallend ist außerdem noch, dass die prozentualen Werte des *Coin-Keepers* und des *Minimalist-Spenders* gleich sind. Das kommt daher, dass beide nach einer Transaktion eine minimale Partition des Betrags in ihrer Geldbörse haben, der *Coin-Keeper* aufgrund von Annahme (3) und der *Minimalist-Spender* per Definition. Der Unterschied ist daher nur die absolute Anzahl an Münzen, nicht jedoch der relative Anteil der einzelnen Werte.

Bei diesem direkten Vergleich mit der U.S. Mint 2014 muss natürlich bedacht werden, dass der Cent-Anteil der Preise als gleichverteilt angenommen wurde, was nicht ganz der Realität entspricht, da viele Preise heutzutage auf 9 Cent enden. Darüber hinaus ist zu berücksichtigen, dass nicht jeder durchgehend eine einheitliche Ausgabestrategie verfolgt.

Literatur

[1] Anna Adamaszek und Michal Adamaszek, *Combinatorics of the change-making problem*, European Journal of Combinatorics **31** (2010) 47–63

[2] Lara Pudwell und Eric Rowland, *What's in YOUR Wallet?* The Mathematical Intelligencer, Volume 31, Number 4, 2015