Die allgemeine Gewinnstrategie des Nim-Spiels geht auf den Harvard-Professor Charles Bouton zurück, siehe The Annals of Mathematics, 2nd Series, Vol 3, 1/4, 1901-02.
Hierzu definieren wir:

4.26 <u>Def</u> (<u>Nim-Addition</u>):

Seien $a, b \in \mathbb{N}_0$. Schreibe a und b binär, also im 2-er System: $a = \sum_{i \geq 0} a_i 2^i$, $b = \sum_{i \geq 0} b_i 2^i$

mit $a_i, b_i \in \{0,1\}$, siehe Thm 4.3 mit $q=2$.

Dann ist
$$a \oplus b := \sum_{i \geq 0} z_i 2^i$$

mit
$$z_i = \begin{cases} 0 & \text{falls } a_i + b_i \text{ gerade,} \\ 1 & \text{falls } a_i + b_i \text{ ungerade.} \end{cases}$$

4.27 <u>Bem</u>: Den Rechenregeln in $(\mathbb{Z}_2, +)$ entnehmen wir:

(a) Nim-Addition ist kommutativ: $a \oplus b = b \oplus a$, $\forall a, b \in \mathbb{N}_0$.

(b) Nim-Addition ist assoziativ: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$, $\forall a, b, c \in \mathbb{N}_0$.

(c) Es ist $a \oplus 0 = a = 0 \oplus a$, $\forall a \in \mathbb{N}_0$.

(d) Es ist $a \oplus a = 0$, $\forall a \in \mathbb{N}_0$.

Damit ist $(\mathbb{N}_0, \oplus)$ eine Gruppe, abelsch, und jedes Element a ist sein eigenes Inverses.

4.28 <u>Bsp</u>: $6 \oplus 15 = (4+2) \oplus (8+4+2+1) = 8+1 = 9$.

denn
$$\begin{array}{rl} 6 = & (1\,1\,0)_2 \\ \oplus\, 15 = & (1\,1\,1\,1)_2 \\ \hline 9 = & (1\,0\,0\,1)_2 \end{array}$$

- Schreibe 6 und 15 im Binärsystem
- Addiere "Bit"weise die Koeffizienten und rechne dabei modulo 2. z.B. $1+1 = 2 = 0 \bmod 2$.
- Rechne Ergebnis um ins Zehnersystem

4.29 Übung:

(a) Erstellen Sie die Additionstafel für die Nim-Addition von $a \oplus b$ mit $a, b \leq 4$.

$\oplus$	1	2	3	4
1				
2				
3				
4				

(b) Bestimmen Sie die Nim-Summe von allen $C = (n_1, n_2, n_3)$ aus Übung 4.25. Hierbei definieren wir die Nim-Summe von C als $n_1 \oplus n_2 \oplus n_3$. Was beobachten Sie?

Diese Beispiele liefern Evidenz für den folgenden Satz:

4.30 Thm:

Der anziehende Spieler im Nim-Spiel $C = (n_1, \dots, n_t)$ hat eine Gewinnstrategie, dh. C ist unsicher, genau dann, wenn $n_1 \oplus \dots \oplus n_t \neq 0$. Die Gewinnstrategie ist dann bei jedem Zug so zu ziehen, daß nach dem Zug die Nim-Summe Null ist.

Beweis:

Wir beweisen diesen Satz in zwei Schritten und zeigen:

(a) Ist die Nim-Summe Null, so ist die Nim-Summe nach dem nächsten Zug immer ungleich Null.

(b) Ist die Nim-Summe ungleich Null, so existiert immer ein Zug, so daß die Nim-Summe danach Null ist.

Ist also am Anfang die Nim Summe $N(C)$ von C ungleich Null, so kann der 1. Spieler nach (b) seinen Zug so machen, daß gilt $N(C_1) = 0$ mit C_1 Konfiguration nach dem 1. Zug. Der 2. Spieler produziert mit seinem beliebigen nächsten Zug eine

Konfiguration C_2 mit $N(C_2) \neq 0$ nach (a).
Induktiv folgt: Nur der 1. Spieler kann (bei optimalem Spiel) nach t Schritten die Konfiguration $C_t = (0, \dots, 0)$ erzeugen, denn $N(C_t) = 0$. Also hat der 1. Spieler eine Gewinnstrategie, falls am Anfang gilt $N(C) \neq 0$. Andernfalls hat der 2. Spieler eine Gewinnstrategie.

4.31 <u>Lemma:</u>
Gegeben sei das Nim Spiel $C = (n_1, \dots, n_t)$ mit $N(C) = 0$. Sei C_1 die Nim-Konfiguration nach dem nächsten Zug.
$\Rightarrow N(C_1) \neq 0$.
(Es ist also egal, welchen Zug wir vornehmen, in jedem Fall erzeugen wir eine Nim-Konfiguration mit Nim-Summe ungleich Null.)

<u>Beweis:</u>
Nach Voraussetzung ist $N(C) = n_1 \oplus \dots \oplus n_t = 0$.
Sei $C_1 = (m_1, \dots, m_t)$ die Konfiguration nach dem nächsten Zug. Dann existiert $1 \leq j \leq t$ mit

$$m_i = \begin{cases} n_i & \text{für } i \neq j, \\ n_i - c & \text{für } i = j, \end{cases}$$

mit $1 \leq c \leq n_i$. Jedes Element in der Gruppe $(\mathbb{N}, \oplus)$ ist selbstinvers.

$$\Rightarrow N(C_1) \overset{4.27}{=} 0 \oplus N(C_1)$$
$$\overset{4.27}{=} \overbrace{N(C) \oplus N(C)} \oplus N(C_1)$$
$$= N(C) \oplus (n_1 \oplus \dots \oplus n_t) \oplus (n_1 \oplus \dots \oplus n_{j-1} \oplus m_j \oplus m_{j+1} \dots)$$
$$\overset{4.27}{=} \underbrace{N(C)}_{=0} \oplus n_j \oplus m_j \overset{\text{Vor.}}{=} n_j \oplus m_j \neq 0,$$

da $n_j \neq m_j$ ist. #

4.32 <u>Lemma</u>:

Gegeben ist das Nim-Spiel $C = (n_1, \dots, n_t)$ mit $N(C) \neq 0$. Dann existiert immer ein Zug derart, daß für die nach diesem Zug erhaltene Nim-Konfiguration C_1 gilt $N(C_1) = 0$.

<u>Beweis</u>: Sei $s := N(C) = n_1 \oplus \dots \oplus n_t \neq 0$.

(a) Schreibe die Binärdarstellungen der Zahlen n_i hin, $1 \leq i \leq t$, und bilde die Nim-Summe $s = (z_r z_{r-1} \dots z_1 z_0)_2$ im Binärsystem. Da $s \neq 0$ existiert $0 \leq k \leq r$ mit $z_k = 1$.

Setze $k := \max\{i \mid z_i = 1\}$.

Nach Konstruktion der Nim-Summe existiert dann ein $1 \leq j \leq t$ mit: der Koeffizient von 2^k in der Binärdarstellung von n_j ist Eins. Von diesem Stapel j entnehmen wir Hölzer. Wieviele Hölzer wir entnehmen, bestimmen wir im nächsten Beweisschritt.

(b) Sei $C_1 = (m_1, \dots, m_t)$. Hierbei ist $m_i = n_i$ für $i \neq j$.

Mit der Rechnung aus dem Beweis von Lemma 4.31 gilt:

$N(C_1) = s \oplus n_j \oplus m_j$. Wir suchen m_j mit $N(C_1) = 0$.

(i) Definiere $m_j := s \oplus n_j$. Dann ist $N(C_1) = s \oplus n_j \oplus s \oplus n_j = 0$.

(ii) Wir müssen zeigen $m_j < n_j$. Wir vergleichen die Binärdarstellungen von m_j und n_j.

Ist $a = \sum_{i \geq 0} z_i 2^i$ in der Binärdarstellung, so bezeichnen wir z_i als die i-te Ziffer von a (nicht übliche Sprachgebrauch).

- Die i-ten Ziffern von n_j und m_j stimmen überein für $i > k$.
- Es ist nach Konstruktion die k-te Ziffer von n_j Eins, aber die k-te Ziffer von m_j Null.

$\Rightarrow m_j < n_j$, (die Ziffern z_i von m_j für $i \leq k-1$ liefern höchstens einen Summanden $2^k - 1$). #

Hiermit ist Thm 4.30 vollständig bewiesen.

Es gibt viele Varianten des klassischen Nim-Spiels. Eine davon ist das Fibonacci-Nim.

4.33 <u>Def</u> (Fibonacci-Nim)

Ein Stapel mit n Hölzern liegt auf dem Tisch. Bei jedem Zug wird mindestens ein Hölzchen entfernt. Außerdem gilt: Beim ersten Zug dürfen maximal $n-1$ Hölzer entfernt werden. Bei jedem weiteren Zug dürfen maximal zweimal soviele Hölzer wie im vorigen Zug entfernt werden.

4.34 <u>Bsp</u>: Seien $n = 16$ Hölzer auf dem Tisch. Angenommen der 1. Spieler entfernt $r_1 = 3$ Hölzer. (Er dürfte maximal $q_1 = 15$ Hölzer entfernen.)

$\Rightarrow$ Der 2. Spieler darf dann maximal $q_2 = 6$ Hölzer entfernen. Angenommen der 2. Spieler nimmt $r_2 = 4$ Hölzer weg.

$\Rightarrow$ Der nun wieder ziehende 1. Spieler darf dann maximal $q_3 = 8$ Hölzer nehmen. Etc.

Wie sieht es mit einer Gewinnstrategie im Fibonacci-Nim aus? Diese läßt sich im Fibonacci-Zahlensystem beschreiben. Wir erinnern an unsere Notation: Sei $a_1 := 1$, $a_2 := 2$ und $a_{n+1} := a_n + a_{n-1}$ für $n \geq 2$. Die Fibonacci-Zahlen sind also

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	...
1	2	3	5	8	13	21	34	55	89	

4.35 <u>Bem</u>: Sei $n = \sum_{i=1}^{t} z_i a_i$ mit $z_i \in \mathbb{N}_0$.

(a) Ist $z_i \in \{0,1\}$ und gilt $z_i z_{i+1} = 0$ für alle $1 \leq i \leq t-1$, die Bedingungen aus Theorem 4.11, so sprechen wir von der FibZ-Darstellung von n.

(b) Ist $z_i \in \mathbb{N}_0$, vielleicht $z_i \in \{0,1\}$, so sprechen wir von einer Fib-Darstellung von n.

Wir schreiben in beiden Fällen wie auch beim g-adischen System $n = (z_t z_{t-1} \cdots z_2 z_1)_f$.

4.36 <u>Bsp</u> Beispielsweise gilt:

$$(4)_{10} = (101)_f$$

$$(5)_{10} = (101)_f + (1)_f = (102)_f$$

$$= \underbrace{(110)_f}_{\text{Fib-Darstellung}} = \underbrace{(1000)_f}_{\text{FibZ-Darstellung}}$$

Beim Umschreiben benutzt man die Rekursion der Fibonacci-Folge $a_{n+1} = a_n + a_{n-1}$. Sinnvollerweise setzen wir noch $a_0 := 1$. Beispielsweise ist

$$(1020)_f = a_4 + 2a_2 = a_4 + a_2 + (a_1 + a_0)$$

$$= a_4 + (a_2 + a_1) + a_0$$

$$= a_4 + a_3 + a_0 = a_5 + a_1$$

$$= (10001)_f$$

bzw $(1020)_f \overset{(a_0 = a_1)}{=} (1012)_f = (1101)_f = (10001)_f$.

<u>Übung</u>: Bestimmen Sie $(1010101000)_f + (100)_f$.

4.37 <u>Notation</u>:
Im Fibonacci-Nim sind die Regeln bei verschiedenen Zügen anders. Wir benötigen entsprechend Notation für die Regeln im m-ten Zug.

Setze $q_1 := n-1$, $q_i := 2v_{i-1}$ für $i \geq 2$, wobei
$v_i :=$ # der im i.ten Zug entnommenen Hölzer.

Sei die Anzahl der Hölzer auf dem Tisch vor dem m-ten Zug $n := n_m = (z_t z_{t-1} \cdots z_2 z_1)_f$ in Fibz-Darstellung.
Definiere $i_m := \min\{i \mid z_i = 1\}$. Dies ist also die Position, der am weitesten rechts stehenden Eins in der Fibz-Darstellung von n.

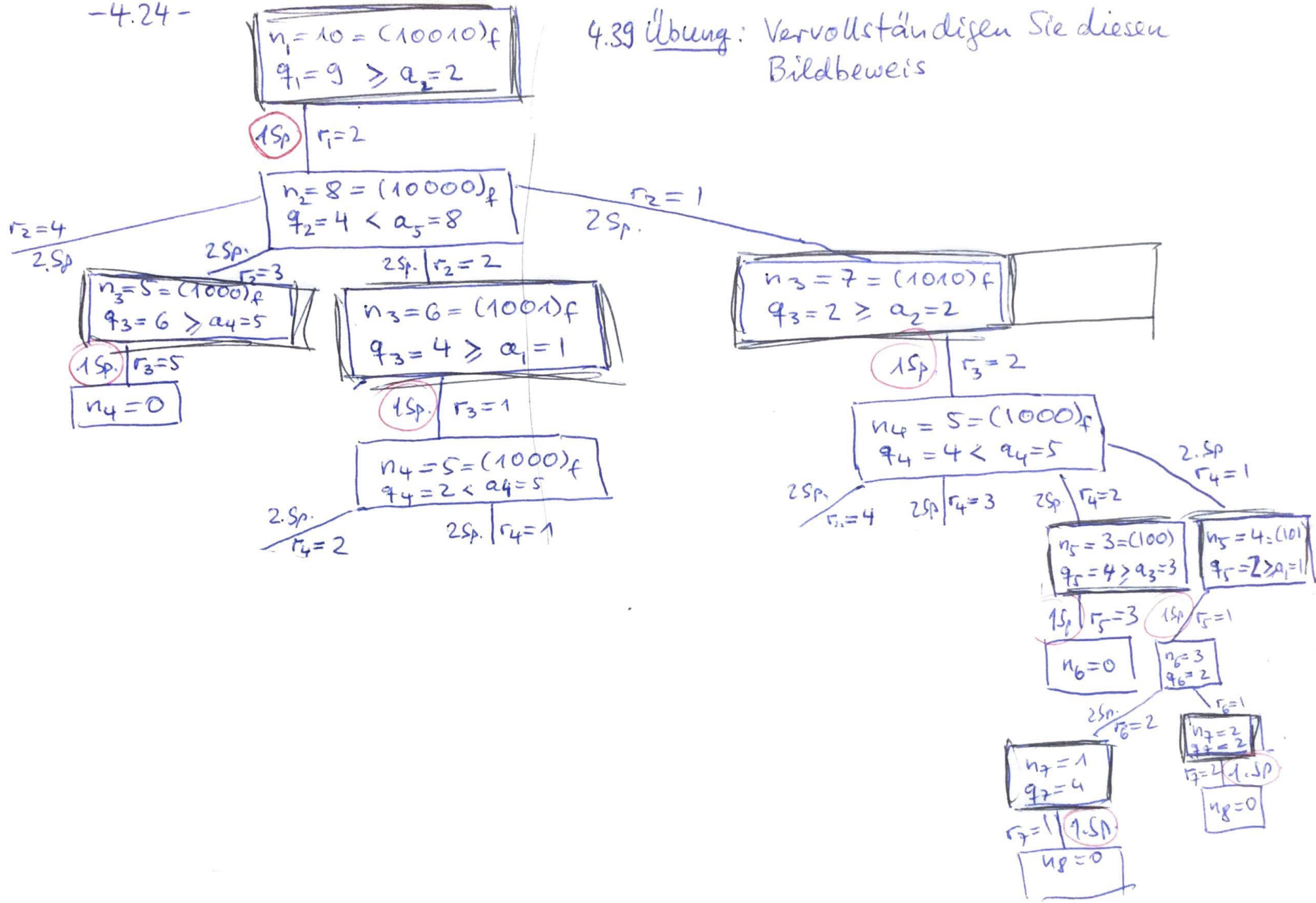
4.38 <u>Theorem</u>:
(a) Ist $q_m < a_{i_m}$ vor dem m.ten Zug, dann ist dies eine sichere Konfiguration.
(b) Ist $q_m \geq a_{i_m}$, dann ist die Konfiguration unsicher, und die Gewinnstrategie besteht aus dem Entfernen von a_{i_m} Hölzern.

4.39 <u>Bsp</u>: (i) Sei $n = 10$, $q_1 = 9$,
Da $n_1 = 10 = (\underset{8}{1}00\underset{2}{1}0)_f \Rightarrow i_1 = 2$, $a_{i_1} = a_2 = 2$.
Da $q_1 = 9 \geq 2 = a_2 = a_{i_1}$, ist die Konfiguration unsicher.
Der anziehende Spieler entfernt $\overset{r_2 :=}{a_2} = 2$ Hölzer.

(ii) Wir erhalten $n_2 = 8$, $q_2 = 2 \cdot 2 = 4$.
Es ist $n_2 = 8 = (10000)_f$. Hier ist $i_2 = 5$, $a_{i_2} = a_5 = 8$
Es $q_2 = 4 < a_{i_2} = 8$, die Konfiguration ist damit nach der zu beweisenden Behauptung sicher.

<u>Übung</u>: Vervollständigen Sie den Beweis in der folgenden Grafik.

4.39 Übung: Vervollständigen Sie diesen Bildbeweis



Beweis für die Gewinnstrategie des anziehenden 1. Spielers für $n = 10$.

4.40 Beweis von 4.38: Wir müssen zeigen:

(a) Jede unsichere Konfiguration kann immer in eine sichere Konfiguration umgewandelt werden;

(b) jede sichere Konfiguration liefert im nächsten Zug, egal wie gezogen wird, immer eine unsichere Konfiguration.

zu(a): Wir nehmen zunächst an, daß $q_m \geq a_{i_m}$ ist, und entnehmen a_{i_m} Hölzer. Falls $a_{i_m} = n$, so liegen nach dem m-ten Zug Null Hölzer auf dem Tisch, das Spiel endet und die Konfiguration vor dem m-ten Zug war unsicher.

Sei also nun $n > a_{i_m}$. Mit der Definition von $i_m =: i$ folgt, es existiert $k \geq i+2$ mit

$$n = (\ldots \underset{\substack{\uparrow \\ \text{Position } k}}{1} \ldots\ldots \underset{\substack{\uparrow \\ \text{Reine zwei aufeinanderfolgende Einsen}}}{0}\ \underset{\substack{\uparrow \\ i}}{1}\ \underbrace{0 \ldots 0}_{\text{Nullen}})_f$$

$$\Rightarrow \quad a_i < a_{i+1} < a_k$$

$$\Rightarrow \quad 2a_i < \underbrace{a_i + a_{i+1}}_{= a_{i+2}} \leq a_k$$

$$\Rightarrow \quad q_{m+1} \overset{\text{Def}}{=} 2a_i \underset{\neq}{<} a_k .$$

Nach Set-up ist im (m+1)ten Schritt $a_{i_{m+1}} = a_k$, also $i_{m+1} = k$, und $q_{m+1} < a_k = a_{i_{m+1}}$.

Also erhalten wir eine neue Konfiguration im (m+1)-ten Schritt, die sicher ist.

zu (b):

(i) Sei jetzt $q_m < a_{i_m}$. Nach Konstruktion ist

$$n = (\underbrace{\ldots}_{\text{definiert die Zahl } c}\, 1\, \underbrace{0 \ldots 0}_{\text{Nuller}})_f = a_i + c \quad \text{mit } c > 0$$

$i_m =: i$-te Position

Es werden weniger als q_m, also weniger als a_i Hölzer entnommen. Die Anzahl der Hölzer $n_{neu} := n - r_m$ ~~ist~~ also

$$n_{neu} = (\underbrace{\ldots\ldots}_{\text{Beitrag } c \text{ ändert sich nicht}}\ \underbrace{\qquad\qquad}_{\text{Beitrag } c_1}\ \underset{\text{Position } h}{1}\ \underbrace{0 \ldots 0}_{\text{Nuller}})_f$$

wobei $i_{m+1} =: h$ die neue Eins ganz rechts in der Fibz-Darstellung von n_{neu} ist.

Es existiert also $c_1 \geq 0$ mit $n_{neu} = c + c_1 + a_h$.

(ii) Mit der Rekursion der Fibonacci-Folge gilt:

Falls $c_1 + a_h = a_{i-1} + a_{i-3} + \ldots + a_{h+2} + a_h$

$\Rightarrow c_1 + a_h + a_{h-1} = a_i$

Bild:

$$\begin{array}{rl} & 1010\ldots1\overset{h\text{-te Position}}{1}00\ldots0 \\ + & \qquad\qquad\qquad 10\ldots0 \\ \hline = & 10101\ldots0110\ldots0 \\ = & 10101\ldots1011000\ldots0 \\ = & \ldots \\ = & 110\ldots\qquad\qquad 0 \\ = & 1000\ldots\qquad\qquad 0. \end{array}$$

(iii) Ist $c_1 + a_h < a_{i-1} + a_{i-2} + \dots + a_{h+2} + a_h$

$\overset{(ii)}{\Longrightarrow} \quad c_1 + a_h + a_{h-1} < a_i$

$\Longrightarrow \quad a_i - c_1 - a_h > a_{h-1}$.

$\Longrightarrow \quad r_m \overset{\text{Def}}{=} n - u_{en} = a_i - (c_1 + a_h) > a_{h-1}$

Wir haben also gezeigt: Werden im m-ten Zug r_m Hölzer entnommen, derart, daß $i_{m+1} = h$ ist, so wurden mindestens a_{h-1} Hölzer entfernt.

$\Longrightarrow \quad q_{m+1} = 2r_m \geq 2a_{h-1} \geq a_{h-1} + a_{h-2} = a_h = a_{i_{m+1}}$,

was zu zeigen war: Egal wie viele Hölzer r_m im m.ten Zug entfernt werden (natürlich mit $r_m \leq q_m$), wir erhalten eine Konfiguration, die unsicher ist.