Vorlesung 11 | 1.12.2020 | 14:15–16:00 via Zoom

(Ein Handzettel für heutige Vorlesung finden Sie auf der Seite der Vorlesung auf meiner Website, siehe Tagebuch)

Prüfungs (in Präsenz): **9/2**, 13/3.

4 Summe unabhängiger Z.V.

(Kapitel 3 und 4 in Bovier Skript)

In der letzten Vorlesungen haben wir gesehen: unendliche Produkten, Summe unabhängiger Z.V., Faltungen, stabilität gegen Faltungen.

Heutige Vorlesung: die Irrfahrt, das Ruinproblem, das Arcsinusgesetz.

4.3 Die Irrfahrt (Random walk)

Wir betrachten nun, was mit sehr großen Summen unabhängiger Zufallsvariablen los ist.

Definition 1. *Sei $X_1, X_2, \dots$ eine Folge von i.i.d. Z.V. mit $\mathbb{P}(X_k = 1) = 1 - \mathbb{P}(X_k = -1) = p$ (Rademacher variablen wenn $p = 1/2$). Dann heißt die Folge*

$$n \mapsto S_n := \sum_{k=1}^{n} X_k$$

die einfache Irrfahrt auf $\mathbb{Z}$ (simple random walk).
Fur $p = 1/2$ heißt die einfache symmetrische Irrfahrt.

Bemerkung.

- $S_n \colon \Omega \to \mathbb{Z}$ ist $\sigma(X_1, \dots, X_n)$ messbar, weil $S_n = f(X_1, \dots, X_n)$. Und $\sigma(X_1, \dots, X_n) = \sigma(S_1, \dots, S_n)$ weil $X_k = S_k - S_{k-1}$.
- $\sigma(X_n)$ und $\sigma(S_1, \dots, S_{n-1})$ unabhängig sind. (Übung)
- Übung:

$$\mathbb{E}[S_n] = n\,\mathbb{E}[X_1], \qquad \mathrm{Var}(S_n) = n\,\mathrm{Var}(X_1).$$

Interpretation. Wettspiel. S_n = Gewinn wenn jedes mal €1 einsetzt und erhält man

$$\begin{cases} €2 \text{ falls die Münze richtig geraten wird;} & S_{n+1} = S_n - 1 + 2 \\ €0 \text{ falls man falsch rät.} & S_{n+1} = S_n - 1 \end{cases}$$

mit gleich W-keit.

Simulationsverfahren

Wir wollen „experimentelle Stochastik“ machen. Die Monte-Carlo-Methode bezieht sich auf die Verwendung von Pseudozufallszahlen, um Stichproben gegebener Zufallsvariablen zu erzeugen. Wir wollen es jetzt verwenden, um die einfache Irrfahr zu generieren.

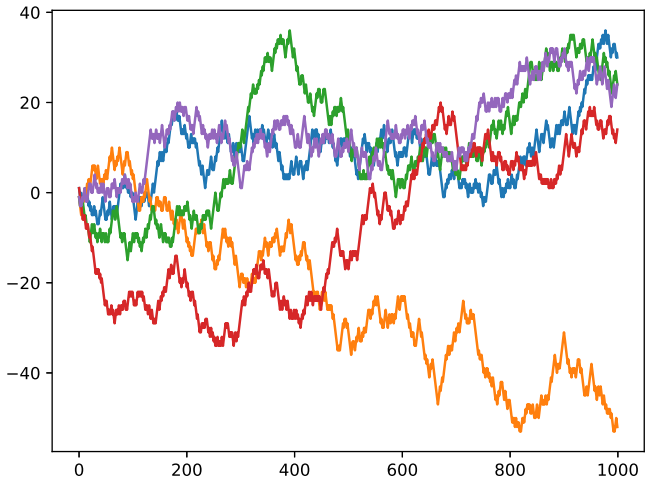
Wir werden die Programmiersprache PYTHON und die MATPLOTLIB Programmbibliothek benutzen, um die Verfahren zu implementieren.

Die Funktion `numpy.random.randint` gibt einen Vektor von pseudozufälligen Ganzzahlen zwischen zwei Grenzen zurück.

```
>>> # 1d random walk.
    import random # to access the random number generator
    import numpy as np # functions for numerical computations
    import matplotlib.pyplot as plt # plotting library

    # probability to move up or down
    prob = [0.5, 0.5]
```

```
>>> t_max = 1000 # maximum time to simulate
    t = np.arange(t_max) # creates the vector 0,1,2,3,...,t_max-1
    # steps can be -1 or 1 (note that randint excludes the upper limit)
    steps = 2 * np.random.randint(0, 1 + 1, t_max) - 1 # X_n
    # the time evolution of the position is obtained by successively
    # summing up individual steps.
    positions = np.cumsum(steps) # S_n
    # plotting down the graph of the random walk
    #plt.clf() # clear the plot
    plt.plot(t,positions)
    fig = plt.gcf() # generate an output file
    ps_out(fig) # ship the output to TeXmacs as a Postscript file
```




```
>>>
```

4.3.1 Das Ruinproblem

Ein Spiel.

- Anfangskapital $K_0 \in \mathbb{N}$.
- Spiel endet falls das Gewinn gleich G ist, oder das ganze Kapital weg ist.

> **Frage:** Was ist die W-keit dass die Spielfolge mit dem Ruin des Spielers endet?

Wie formalisieren das?

- Die Anzahl der Spielrunden ist nicht im voraus bekannt $\Rightarrow \infty$-Produktraum ist nötig $\Omega = \{-1,1\}^{\mathbb{N}}$, $\mathcal{F} = \sigma(X_k : k \geqslant 1)$, $\mathbb{P} = \mu^{\otimes \mathbb{N}}$ mit $X_k(\omega) = \omega_k$ und $\mu(\{-1\}) = \mu(\{1\}) = 1/2$. (Das ist der kleinste W-raum wir brauchen)
- Sei K_n das Kapital nach n Spielrunden.

  $$K_n := K_0 + S_n$$

  mit $p = 1/2$.
- Das Spiel endet mit Ruine nach n Runden:

  $$A_n = \{S_n = -K_0\} \cap \bigcap_{k=1}^{n-1} \{-K_0 < S_k < G\}$$

- Gesucht ist die W-keit, dass wir irgendwann ruiniert werden:

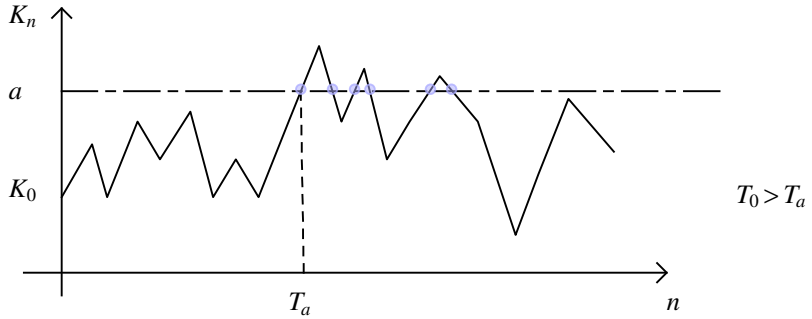
$$\mathbb{P}(A) \qquad \text{mit } A = \bigcup_{n\geqslant 1} A_n = \{\exists n \geqslant 1 : S_n = -K_0\} = \{\exists n \geqslant 1 : S_n \leqslant -K_0\}.$$

- *Stoppzeiten*: Sei T_a die erste Zeitpunkt n s.d. die Kapital das Niveau $a \in \mathbb{Z}$ erreichen:

$$T_a = \inf\{n \geqslant 1 : K_n = a\} \qquad \left[\inf \emptyset = +\infty\right]$$

$$T_a : \Omega \to \mathbb{N} \cup \{+\infty\}$$

ist ein Funktion über Ω dass das hängt von allen (unednliche viele Z.V.) $X_1, X_2, \ldots$ ab. Wir können es nicht im endliche Produktraüm $\{-1,1\}^N$ darstellen/kostruire. Wir müssen auf $\{-1,1\}^{\mathbb{N}}$ arbeiten.



$$A = \left\{\inf\{n \geqslant 1 : S_n = -K_0\} < \inf\{n \geqslant 1 : S_n = G\}\right\} = \{T_0 < T_{\tilde{G}}\}$$

mit $\tilde{G} = K_0 + G$.

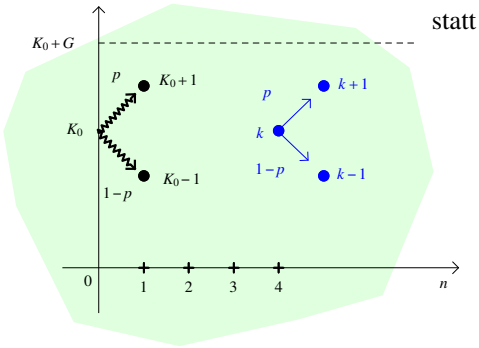
- Sei $h : \{0, \ldots, \tilde{G}\} \to [0,1]$ so dass

$$\begin{cases} h(k) := \mathbb{P}\left(T_0 < T_{\tilde{G}} \middle| K_0 = k\right) & \text{für } 0 < k < \tilde{G} \\ h(0) = 1, \\ h(\tilde{G}) = 0. \end{cases}$$

Dann $\mathbb{P}(A) = h(K_0)$ mit $\tilde{G} =: K_0 + G$ und $p = 1/2$.

$h(k)$ ist die Wahrscheinlichkeit, dass $(K_n)_{n\geqslant 1}$ die Niveau 0 vor $\tilde{G}$ erreicht, vorausgesetzt, es beginnt bei K_0 (und dass in jedem Schritt von ein Punkt mit den Wahrscheinlichkeiten p und $1-p$ angehoben oder abgesenkt wird)

<u>Ziel</u>: Finde $h(k)$.



Nach 1 Schritt, falls $K_1 \notin \{0, K_0 + G\}$ dann wie zuvor aber mit K_1 statt K_0!

Für $0 < k < \tilde{G}$:

$$h(k) = (1-p)\mathbb{1}_{k=1} + \mathbb{1}_{k=1} p h(2) + \mathbb{1}_{k>1}(1-p)h(k-1) + \mathbb{1}_{k>1} p\left(\mathbb{1}_{k+1=\tilde{G}} \cdot 0 + \mathbb{1}_{k+1<\tilde{G}} h(k+1)\right)$$

$$h(k) = (1-p)h(k-1) + p h(k+1)$$

Wir müssen lösen

$$\begin{cases} h(k) &= (1-p)h(k-1)+ph(k+1), \quad 0<k<\tilde{G} \\ h(0) &= 1 \\ h(\tilde{G}) &= 0 \end{cases} \tag{1}$$

$h(0), h(\tilde{G})$ sind Randwerte. Stellt (1) eine *diskretes Randwertaufgabe* dar. In Analogie zu der entsprechenden Differentialgleichung wird sia auch als *Dirichletproblem* bezeichnet.

Lösung für $p = 1/2$ (Siehe Blatt 6)

$$h(k) = 1 - \frac{k}{\tilde{G}}.$$

Dann

$$\mathbb{P}(A) = \frac{G}{G+K_0}$$

Insbesonderes, wenn $G \to \infty$,

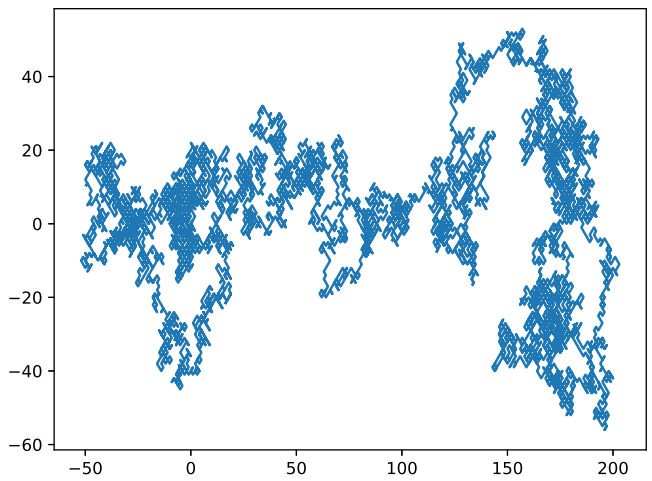
$$\mathbb{P}(A) \to 1,$$

d.h. wenn wir weiter spielen werden wir mit W-keit 1 das ganze Kapital verlieren. **(Ruin ist sicher!)**

Das bedeutet auch dass ist sicher dass die Irrfahr das Niveau 0 früher oder später erreichen.

(Das ist nich Wahr wenn die Irrfahr mehreredimensionalen ist.)

```
>>>
>>> # a 2d random walk (note: this is *not* the simple 2d random walk).
    t_max = 10000 # maximum time to simulate
    t = np.arange(t_max) # creates the vector 0,1,2,3,...,t_max-1
    # steps can be -1 or 1 (note that randint excludes the upper limit)
    steps_x = 2 * np.random.randint(0, 1 + 1, t_max) - 1 # X_n
    steps_y = 2 * np.random.randint(0, 1 + 1, t_max) - 1
    # the time evolution of the position is obtained by successively
    # summing up individual steps.
    positions_x = np.cumsum(steps_x) # S_n
    positions_y = np.cumsum(steps_y)
    # plotting down the graph of the random walk
    plt.clf() # clear the plot
    plt.plot(positions_x,positions_y)
    fig = plt.gcf() # generate an output file
    ps_out(fig) # ship the output to TeXmacs as a Postscript file
```



>>>

4.3.2 Das Arcsinusgesetz

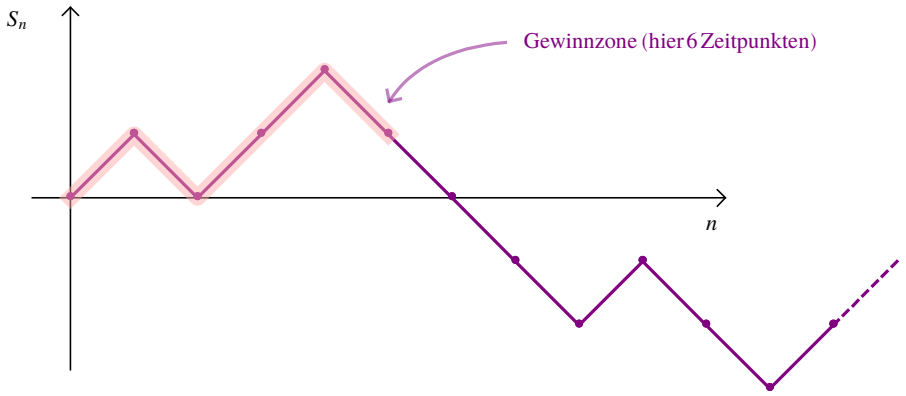
Sei (symmetrische einfache Irrfahrt)

$$S_n = \sum_{k=1}^{n} X_k, \qquad S_0 = 0,$$

mit $\mathbb{P}(X_k = 1) = \mathbb{P}(X_k = -1) = 1/2$ (Rademacher variablen) und $(X_n)_{n\geqslant 1}$ Folge von iid Z.V.

Frage: Wie viel Zeit verbringt es in der Gewinnzone?

Gewinnzone: die Menge der Zeitpunk k mit $S_k > 0$ oder $S_{k+1} > 0$.



Wir wollen die folgenden Zufallsvariablen analysieren.

(a). $Z_{2n} := \sum_{\ell=1}^{2n} Y_\ell$ mit

$$Y_\ell = \begin{cases} 1, & \text{falls } S_\ell > 0 \text{ oder } S_{\ell+1} > 0, \\ 0, & \text{sonst.} \end{cases}$$

$$\Rightarrow Z_{2n} = \sharp\text{schritten in positiven Bereich}$$

(b). $f_{2n} := \mathbb{P}\Big(\underbrace{\inf\{\ell > 0 : S_\ell = 0\}}_{\text{erste Rückkehr nach 0}} = 2n\Big)$

(c). Die W-keit Rückkehr nach $2n$ Schritten

$$u_{2n} := \mathbb{P}(S_{2n} = 0)$$

Lemma 2. *Es gilt*

$$(a) \quad u_{2n} = \frac{1}{2^n}\binom{2n}{n}$$

$$(b) \quad f_{2n} = \frac{1}{2^n} u_{2n-2} = u_{2n-2} - u_{2n}$$

Beweis. (a) Einfach, Binomialverteilung:

$$\frac{S_{2n} + 2n}{2} \sim \mathrm{Bin}\Big(2n, \frac{1}{2}\Big) \Rightarrow S_{2n} = 0 \Leftrightarrow \mathrm{Bin}\Big(2n, \frac{1}{2}\Big) = n \Leftrightarrow p = \Big(\frac{1}{2}\Big)^n \Big(\frac{1}{2}\Big)^{2n-n} \binom{2n}{n}$$

(b) Sei

$$g_{2n} := \mathbb{P}(S_{2n} = 0, S_k > 0 \text{ für } k = 1, \ldots, 2n-1)$$

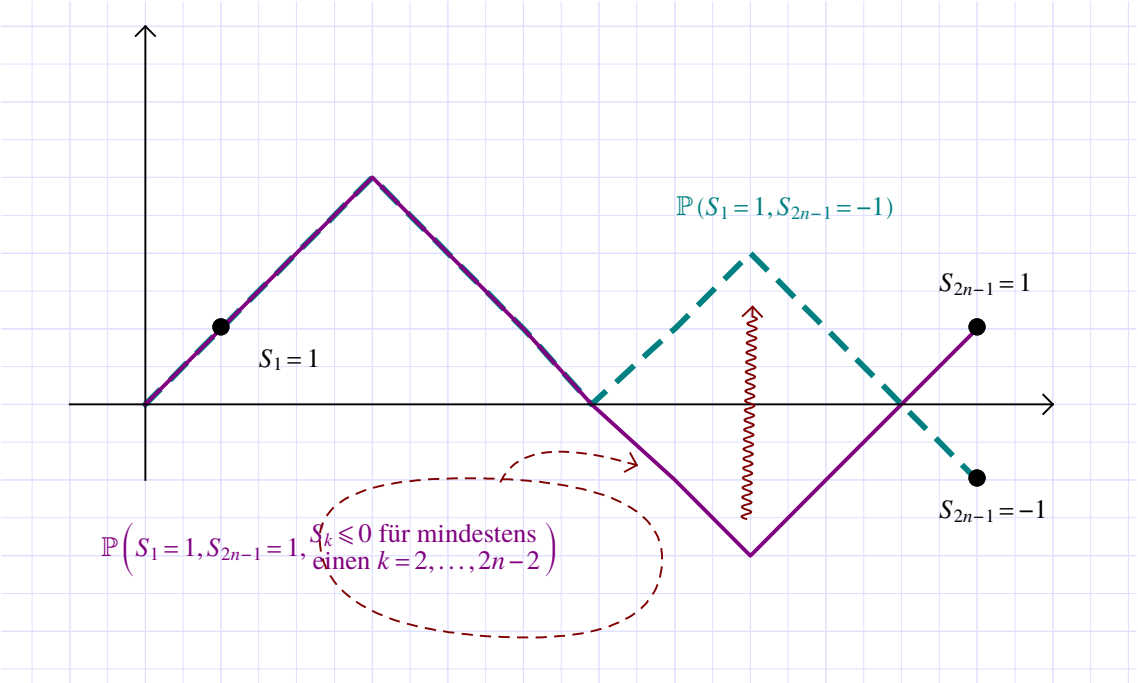
Dann $f_{2n} = 2g_{2n}$ weil für f_{2n} können die Pfade >0 oder <0 für $k = 1, \ldots, 2n-1$ sein.

$$g_{2n} = \underbrace{\mathbb{P}(S_{2n} = 0 | S_{2n-1} = 1)}_{=1/2} \mathbb{P}(S_{2n-1} = 1, S_k > 0, k = 1, \ldots, 2n-2)$$

$$= \frac{1}{2}\left[\mathbb{P}(S_1 = 1, S_{2n-1} = 1) - \underbrace{\mathbb{P}\left(S_1 = 1, S_{2n-1} = 1, \begin{matrix} S_k \leqslant 0 \text{ für mindestens} \\ \text{einen } k = 2, \ldots, 2n-2 \end{matrix}\right)}_{=\mathbb{P}(S_1 = 1, S_{2n-1} = -1)}\right]$$

Reflexionprinzip (Reflection principle):

Für jeden Pfad (Violett) erstellen wir einen anderen (Grün), der die gleichen Entscheidungen trifft, bis er beim ersten Mal Null erreicht, und dann genau die entgegengesetzten Entscheidungen des ursprünglichen Pfades trifft. Wenn also der erste Pfad in a endet, endet der zweite in $-a$. Sie haben die gleiche Wahrscheinlichkeit.



Dann

$$f_{2n} = 2\,g_{2n} = \mathbb{P}(S_1 = 1, S_{2n-1} = 1) - \mathbb{P}(S_1 = 1, S_{2n-1} = -1)$$

$$= \frac{1}{2}\frac{1}{2^{2n-2}}\binom{2n-2}{n-1} - \frac{1}{2}\frac{1}{2^{2n-2}}\binom{2n-2}{n} = \underbrace{\left[\frac{\binom{2n-2}{n-1}}{2^{2n-2}}\right]}_{=u_{2n-2}}\frac{1}{2n}$$

Aber auch

$$\frac{1}{2^{2n}}\binom{2n}{n} = \binom{2n-2}{n-1}\frac{1}{2^{2n-2}} \cdot \frac{2n(2n-1)}{n \cdot n \cdot 2 \cdot 2} = \binom{2n-2}{n-1}\frac{1}{2^{2n-2}} \cdot \frac{2n-1}{2} = 1 - \frac{1}{2n}.$$

Dann

$$u_{2n} = u_{2n-2} - f_{2n}.$$

□

Jetz wollen wir Z_{2n} bestimmen und das $n \to \infty$ asymptotische Verhältnis berechnen.

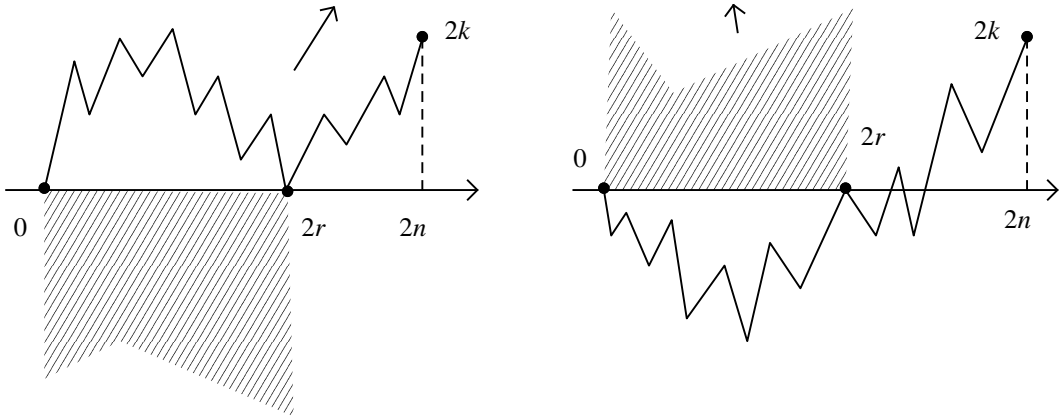
Satz 3. *Es gilt*

$$\mathbb{P}(Z_{2n} = 2k) =: p_{2k,2n} = u_{2k} \cdot u_{2n-2k} = \binom{2k}{k} \frac{1}{2^{2k}} \binom{2n-2k}{n-k} \frac{1}{2^{2(n-k)}}$$

Erinnerung $u_{2n} = \mathbb{P}(S_{2n} = 0)$.

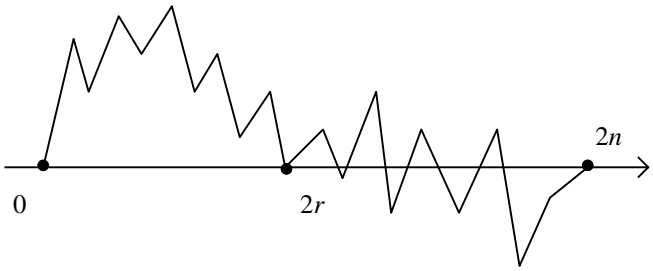
Beweis. Es gilt

$$p_{2k,2n} = \frac{1}{2} \sum_{r=1}^{n} f_{2r} \cdot p_{2k-2r,2n-2r} + \frac{1}{2} \sum_{r=1}^{n-k} f_{2r} \cdot p_{2k,2n-2r}$$



Das ist eine Zerlegung von Ω nach dem ersten Mal, wenn der Irrfahrt Null erreicht.
Wir versuchen die Rekursion lösen, ohne zunächts f_{2r} zu berechnen. Dazu bemerken wir, dass

$$u_{2n} := \mathbb{P}(S_{2n} = 0) = \sum_{r=1}^{n} f_{2r} \cdot \underbrace{u_{2n-2r}}_{= \mathbb{P}(S_{2n-2r} = 0)}$$



Zu zeigen:

$$p_{2k,2n} = u_{2k} \cdot u_{2n-2k}.$$

Per Induktion auf k.
(1) $k = 0$: $p_{0,2n} = p_{2n,2n} = u_{2n}$ wegen Symmetrie.

$$\mathbb{P}(S_k > 0, k = 1, \ldots, 2n+1) = \underbrace{\frac{1}{2}}_{= \mathbb{P}(S_1 = 1)} \quad \mathbb{P}(S_k \geqslant 0, 1 \leqslant k \leqslant 2n).$$

$$\Rightarrow p_{2n,2n} = \mathbb{P}(S_k \geqslant 0, 1 \leqslant k \leqslant 2n) = 2\mathbb{P}(S_k > 0, 1 \leqslant k \leqslant 2n)$$

S_{2n+1} kann nicht 0 sein.

$$= \mathbb{P}(\inf\{r > 1 : S_{2r} = 0\} > 2n) = 1 - \sum_{r=1}^{n} f_{2r}$$

$$\underset{\text{Lemma 2}}{=} 1 - \sum_{r=1}^{n} (u_{2r-2} - u_{2r}) = 1 - (u_0 - u_{2n}) = u_{2n}$$

weil $u_0 = 1$.

(2) Induktion über k: Sei

$$p_{2n,2m} = u_{2k} \cdot u_{2m-2k}$$

für $m \leqslant n-1$ und $0 \leqslant k \leqslant m$. Für $n = 1$ gilt $p_{0,0} = 1 = u_0 \cdot u_0$ (OK).

Dann für $m = \tilde{n}$

$$p_{0,2\tilde{n}} = u_{2\tilde{n}} \cdot u_0 = p_{2\tilde{n},2\tilde{n}} \quad \text{Teil (1)}$$

und für $1 \leqslant k \leqslant \tilde{n} - 1$

$$p_{2k,2\tilde{n}} = \frac{1}{2}\left(\sum_{r=1}^{k} f_{2r} \cdot u_{2k-2r}\right) u_{2\tilde{n}-2k}$$

$$+ \underbrace{\frac{1}{2}\left(\sum_{r=1}^{\tilde{n}-k} f_{2r} \cdot u_{2\tilde{n}-2k-2r}\right)}_{= u_{2\tilde{n}-2k}} u_{2k} = u_{2k} \cdot u_{2\tilde{n}-2k}$$

□

Wir wollen die Limes $n \to \infty$ nehmen und die Asymptotischer Verteilung von Z_{2n} berechnen.
