§4 Simulation (Beispiele)

In Kapitel 3 haben wir gesehen, dass sich mit der (durch Simulation) experimentell gewonnenen relativen Häufigkeit $\mathbb{H}[A]$ die Wahrscheinlichkeit $\mathbb{P}[A]$ eines Ereignisses A näherungsweise bestimmen lässt. Dazu muss man das zur Diskussion stehende Zufallsexperiment oft simulieren, aus der Liste der dabei erzeugten Realisierungen mit Hilfe von Mathematica alle jene Realisierungen auswählen, welche diesem Ereignis A entsprechen und schließlich den Quotient aus der Anzahl dieser Realisierungen und der Anzahl der Wiederholungen bilden.

Wir werden dazu zuerst ausführlich darauf eingehen, wie man mit Hilfe von Mathematica aus einer Liste alle jene Elemente auswählen kann, die einer bestimmten Bedingung genügen bzw ein bestimmtes Muster besitzen. Anschließend werden wir an Hand von zahlreichen Beispielen zeigen, wie sich die Wahrscheinlichkeit von Ereignissen mit Hilfe von Simulation näherungsweise bestimmen lässt.

4.1 Auswahl von Elementen aus einer Liste

Wir klären zunächst, was *Mathematica* unter einer Bedingung bzw unter einem Muster versteht. Anschließend zeigen wir, wie man mit Hilfe von *Mathematica* aus einer gegebenen Liste Ω alle jene Elemente ω auswählen kann, die einer gewissen Bedingung genügen bzw die ein bestimmtes Muster besitzen.

4.1.1 Definition: Eine Bedingung wird im Rahmen von *Mathematica* stets in der Form

bedingung[♯] &

angegeben. Die Raute ♯ steht dabei für die Laufvariable, welche die Objekte der Liste durchläuft; das &-Zeichen gibt an, dass die Bedingung als reine Funktion angegeben wird (die Zeichen ♯ und & gehören dabei stets zusammen); bedingung steht für die Bedingung, welche erfüllt sein muss.

▼

4.1.2 Definition: Ein Muster wird im Rahmen von *Mathematica* stets in der Form

muster[*x* _, *y* _, …]

angegeben. Jedes der Zeichen *x* _, *y* _, … ist dabei ein Platzhalter für ein Objekt; muster beschreibt das gesuchte Muster. Muster können auch mit Nebenbedingung versehen werden. Ein mit einer Nebenbedingung *nebenbedingung* versehenes Muster hat dabei die Gestalt

muster[*x* _, *y* _, …] /; *nebenbedingung*

▼

Die Auswahl von Objekten aus einer Liste, welche einer bestimmten Bedingung genügen, bzw die ein bestimmtes Muster besitzen, erfolgt im Rahmen von *Mathematica* durch die Befehle Select, Cases bzw DeleteCases:

■ Select[*liste*, *bedingung*]

wählt aus der Liste *liste* alle jene Objekte aus, welche der Bedingung *bedingung* genügen.

■ Cases[*liste*, *muster*]

wählt aus der Liste *liste* alle jene Objekte aus, welche das Muster *muster* besitzen.

■ DeleteCases[*liste*, *muster*]

löscht aus der Liste *liste* alle jene Objekte, welche das Muster *muster* besitzen.

Wir erläutern die Verwendung dieser Befehle an einer Reihe von typischen Beispielen:

4.1.3 Beispiel: Gesucht ist eine Liste aller jener Punkte $\{x, y\}$ mit $x, y \in \{1, 2, \ldots, 50\}$, welche auf der Geraden $3\,x - 2\,y = 5$ liegen.

▼

Lösung: Wir erzeugen unter Verwendung von Tuples und Range die Liste $\Omega = \{\{x, y\} \mid x, y \in \{1, 2, \ldots, 50\}\}$ und wählen aus dieser Liste die gesuchte Liste mittels Select und der Bedingung 3 #[[1]] – 2 #[[2]] == 5 & aus:

```
Ω = Tuples[Range[50], 2];
Select[Ω, 3 #[[1]] − 2 #[[2]] == 5 &]
Clear[Ω]
```

```
{{3, 2}, {5, 5}, {7, 8}, {9, 11}, {11, 14}, {13, 17}, {15, 20}, {17, 23}, {19, 26},
 {21, 29}, {23, 32}, {25, 35}, {27, 38}, {29, 41}, {31, 44}, {33, 47}, {35, 50}}
```

4.1.4 Beispiel: Gesucht ist eine Liste aller pythagoreischen Tripel $\{x, y, z\}$ mit $x, y, z \in \{1, 2, \ldots, 50\}$. (Unter einem pythagoreischen Tripel versteht man drei Zahlen $x, y, z \in \mathbb{N}$ mit der Eigenschaft $x^2 + y^2 = z^2$.)

▼

Lösung: Analog zu Beispiel 4.1.3 erzeugen wir zuerst die Liste $\Omega = \{\{x, y, z\} \mid x, y, z \in \{1, 2, \ldots, 50\}\}$ und wählen aus dieser Liste die gesuchte Liste mittels Select und der Bedingung #[[1]]^2 + #[[2]]^2 == #[[3]]^2 & aus:

```
Ω = Tuples[Range[50], 3];
Select[Ω, #[[1]]^2 + #[[2]]^2 == #[[3]]^2 &]
Clear[Ω]
```

```
{{3, 4, 5}, {4, 3, 5}, {5, 12, 13}, {6, 8, 10}, {7, 24, 25}, {8, 6, 10}, {8, 15, 17},
 {9, 12, 15}, {9, 40, 41}, {10, 24, 26}, {12, 5, 13}, {12, 9, 15}, {12, 16, 20},
 {12, 35, 37}, {14, 48, 50}, {15, 8, 17}, {15, 20, 25}, {15, 36, 39}, {16, 12, 20},
 {16, 30, 34}, {18, 24, 30}, {20, 15, 25}, {20, 21, 29}, {21, 20, 29},
 {21, 28, 35}, {24, 7, 25}, {24, 10, 26}, {24, 18, 30}, {24, 32, 40}, {27, 36, 45},
 {28, 21, 35}, {30, 16, 34}, {30, 40, 50}, {32, 24, 40}, {35, 12, 37},
 {36, 15, 39}, {36, 27, 45}, {40, 9, 41}, {40, 30, 50}, {48, 14, 50}}
```

4.1.5 Beispiel: Aus der Liste der ersten 100 Paare von aufeinanderfolgenden Primzahlen sind alle Primzahl-Zwillinge auszuwählen. (Unter einem Primzahl-Zwilling versteht man zwei aufeinanderfolgende Primzahlen, deren Differenz gleich 2 ist.)

▼

Lösung: Mit Hilfe von Table und Prime (damit lässt sich die i-te Primzahl aufrufen) erzeugen wir zuerst die Liste Ω der ersten 100 Paare von aufeinanderfolgenden Primzahlen. Aus dieser Liste wählen wir die gesuchte Liste mit Hilfe Select und der Bedingung #[[2]] – #[[1]] == 2 & aus:

```
Ω = Table[{Prime[i], Prime[i + 1]}, {i, 1, 100}];
Select[Ω, #[[2]] − #[[1]] == 2 &]
Clear[Ω]
```

```
{{3, 5}, {5, 7}, {11, 13}, {17, 19}, {29, 31}, {41, 43}, {59, 61},
 {71, 73}, {101, 103}, {107, 109}, {137, 139}, {149, 151}, {179, 181},
 {191, 193}, {197, 199}, {227, 229}, {239, 241}, {269, 271}, {281, 283},
 {311, 313}, {347, 349}, {419, 421}, {431, 433}, {461, 463}, {521, 523}}
```

4.1.6 Beispiel: Aus den Listen der Form $\{x, y, z\}$ mit $x, y, z \in \{1, 2, 3, 4\}$ sind alle jene Listen auszuwählen, bei denen die drei Elemente paarweise verschieden sind.

▼

Lösung: Analog zu Beispiel 4.1.4 erzeugen wir zuerst die Liste $\Omega = \{\{x, y, z\} \mid x, y, z \in \{1, 2, 3, 4\}\}$ und wählen aus dieser Liste die gesuchte Liste mit Select und der Bedingung #[[1]] ≠ #[[2]] ≠ #[[3]] & aus:

```
Ω = Tuples[Range[4], 3];
Select[Ω, #[[1]] ≠ #[[2]] ≠ #[[3]] &]
Clear[Ω]
```

```
{{1, 2, 3}, {1, 2, 4}, {1, 3, 2}, {1, 3, 4}, {1, 4, 2}, {1, 4, 3}, {2, 1, 3}, {2, 1, 4},
 {2, 3, 1}, {2, 3, 4}, {2, 4, 1}, {2, 4, 3}, {3, 1, 2}, {3, 1, 4}, {3, 2, 1}, {3, 2, 4},
 {3, 4, 1}, {3, 4, 2}, {4, 1, 2}, {4, 1, 3}, {4, 2, 1}, {4, 2, 3}, {4, 3, 1}, {4, 3, 2}}
```

4.1.7 Beispiel: Aus allen Listen der Form $\{x, y, z\}$ mit $x, y, z \in \{1, 2, 3, 4\}$ sind alle jene Listen auszuwählen, bei denen $x \neq y$ und $y \neq z$ ist (man beachte den Unterschied zu Beispiel 4.1.6).

▼

Lösung: Analog zu Beispiel 4.1.4 erzeugen wir zuerst die Liste $\Omega = \{\{x, y, z\} \mid x, y, z \in \{1, 2, 3, 4\}\}$ und wählen aus dieser Liste die gesuchte Liste mit Select und der Bedingung #[[1]] ≠ #[[2]] ⋀ #[[2]] ≠ #[[3]] & aus:

```
Ω = Tuples[Range[4], 3];
Select[Ω, #[[1]] ≠ #[[2]] ⋀ #[[2]] ≠ #[[3]] &]
Clear[Ω]
```

```
{{1, 2, 1}, {1, 2, 3}, {1, 2, 4}, {1, 3, 1}, {1, 3, 2}, {1, 3, 4}, {1, 4, 1}, {1, 4, 2},
 {1, 4, 3}, {2, 1, 2}, {2, 1, 3}, {2, 1, 4}, {2, 3, 1}, {2, 3, 2}, {2, 3, 4},
 {2, 4, 1}, {2, 4, 2}, {2, 4, 3}, {3, 1, 2}, {3, 1, 3}, {3, 1, 4}, {3, 2, 1},
 {3, 2, 3}, {3, 2, 4}, {3, 4, 1}, {3, 4, 2}, {3, 4, 3}, {4, 1, 2}, {4, 1, 3},
 {4, 1, 4}, {4, 2, 1}, {4, 2, 3}, {4, 2, 4}, {4, 3, 1}, {4, 3, 2}, {4, 3, 4}}
```

4.1.8 Beispiel: Einer Urne mit $n = 8$ nummerierten Kugeln werden auf einen Griff zufällig $k = 5$ Kugeln entnommen. Gesucht ist eine Liste aller jener Realisierungen, bei denen jedenfalls die Kugeln mit den Nummern 3, 6 und 8 gezogen werden.

▼

Lösung: Wir geben zwei mögliche Lösungen für diese Fragestellung an:

1) Die Menge aller Realisierungen beim gleichzeitigen Ziehen von 5 Kugeln aus einer Urne mit 8 nummerierten Kugeln lässt sich durch die Menge

$$\Omega = \{\{x_1, x_2, \ldots, x_8\} \mid x_1, x_2, \ldots, x_8 \in \{0, 1\} \text{ und } x_1 + x_2 + \ldots + x_8 = 5\}$$

beschreiben, wobei die Liste $\{x_1, x_2, \ldots, x_8\} \in \Omega$ der Realisierung "die i-te Kugel wird genau dann gezogen, wenn $x_i = 1$ ist" entspricht. Diese Menge Ω erzeugen wir, indem wir zuerst mit Tuples und Range die Menge

$$\Psi = \{\{x_1, x_2, \ldots, x_8\} \mid x_1, x_2, \ldots, x_8 \in \{0, 1\}\}$$

erzeugen und auf diese Menge den Befehl Select mit der Bedingung Sum[#[[i]], {i, 1, 8}] == 5 & anwenden. Die von uns gesuchte Liste

$$A = \{\{x_1, x_2, \ldots, x_8\} \in \Omega \mid x_3 = 1 \text{ und } x_6 = 1 \text{ und } x_8 = 1\}$$

erhalten wir dann dadurch, indem wir aus der Liste Ω mit Hilfe von Cases alle jene Elemente auswählen, welche das Muster {x1_, x2_, 1, x4_, x5_, 1, x7_, 1} besitzen:

```
Ψ = Tuples[Range[2] - 1, 8];
Ω = Select[Ψ, Sum[#[[i]], {i, 1, 8}] == 5 &];
Cases[Ω, {x1_, x2_, 1, x4_, x5_, 1, x7_, 1}]
Clear[Ψ, Ω]
```

```
{{0, 0, 1, 0, 1, 1, 1, 1}, {0, 0, 1, 1, 0, 1, 1, 1},
 {0, 0, 1, 1, 1, 1, 0, 1}, {0, 1, 1, 0, 0, 1, 1, 1},
 {0, 1, 1, 0, 1, 1, 0, 1}, {0, 1, 1, 1, 0, 1, 0, 1}, {1, 0, 1, 0, 0, 1, 1, 1},
 {1, 0, 1, 0, 1, 1, 0, 1}, {1, 0, 1, 1, 0, 1, 0, 1}, {1, 1, 1, 0, 0, 1, 0, 1}}
```

2) Die Menge aller Realisierungen beim gleichzeitigen Ziehen von 5 Kugeln aus einer Urne mit 8 nummerierten Kugeln lässt sich aber auch durch die Menge

$$\Omega = \{\{x_1, x_2, \ldots, x_5\} \mid x_1, x_2, \ldots, x_5 \in \{1, 2, \ldots, 8\} \text{ und } x_1 < x_2 < \ldots < x_5\}$$

beschreiben, wobei die Liste $\{x_1, x_2, \ldots, x_5\} \in \Omega$ der Realisierung "es werden die Kugeln mit den Nummern $x_1, x_2, \ldots, x_5$ gezogen" entspricht. Diese Menge Ω erzeugen wir, indem wir zuerst mit Tuples und Range die Menge

$$\Psi = \{\{x_1, x_2, \ldots, x_5\} \mid x_1, x_2, \ldots, x_5 \in \{1, 2, \ldots, 8\}\}$$

erzeugen und darauf den Befehl Select mit der Bedingung #[[1]] < #[[2]] < … < #[[5]] & anwenden. Die von uns gesuchte Liste

$$A = \{\{x_1, x_2, \ldots, x_5\} \in \Omega \mid \{3, 6, 8\} \subseteq \{x_1, x_2, \ldots, x_5\}\}$$

erhalten wir dann offenbar dadurch, indem wir aus dieser Liste Ω mit Hilfe von Select und der Bedingung (¬ FreeQ[#, 3] && ¬ FreeQ[#, 6] && ¬ FreeQ[#, 8]) & alle jene Listen auswählen, welche sowohl die Zahl 3 als auch die Zahl 6 als auch die Zahl 8 enthalten (mit ¬ FreeQ wird geprüft, ob eine Liste ein gewisses Element enthält):

```
Ψ = Tuples[Range[8], {5}];
Ω = Select[Ψ, #[[1]] < #[[2]] < #[[3]] < #[[4]] < #[[5]] &];
Select[Ω, (¬ FreeQ[#, 3] && ¬ FreeQ[#, 6] && ¬ FreeQ[#, 8]) &]
Clear[Ψ, Ω]
```

```
{{1, 2, 3, 6, 8}, {1, 3, 4, 6, 8}, {1, 3, 5, 6, 8}, {1, 3, 6, 7, 8}, {2, 3, 4, 6, 8},
 {2, 3, 5, 6, 8}, {2, 3, 6, 7, 8}, {3, 4, 5, 6, 8}, {3, 4, 6, 7, 8}, {3, 5, 6, 7, 8}}
```

4.1.9 Beispiel: Mit acht Karten, auf denen je ein Buchstabe aufgetragen ist, lässt sich das Wort SEEREISE bilden. Die Karten werden gemischt und dann einzeln gezogen. Man liste alle jene Realisierungen auf, bei denen die ersten vier Buchstaben in der Reihenfolge ihrer Ziehung das Wort REIS bilden?

▼

Lösung: Die Menge aller Anordnungen der Buchstaben des Wortes SEEREISE lässt sich durch die Menge

$$\Omega = \{\{x_1, x_2, \ldots, x_8\} \mid \{x_1, x_2, \ldots, x_8\} \text{ ist eine mögliche Anordnung der Buchstaben } s, e, e, r, e, i, s, e\}$$

beschreiben. Für die Liste aller jener Realisierungen, bei denen die ersten vier Buchstaben in der Reihenfolge ihrer Ziehung das Wort REIS bilden, gilt damit

$$A = \{\{x_1, x_2, \ldots, x_8\} \in \Omega \mid x_1 = r, x_2 = e, x_3 = i, x_4 = s\}$$

Diese Menge A lässt sich erzeugen, indem wir zuerst mit Hilfe von Permutations die Menge Ω, also alle möglichen Anordnungen der acht Buchstaben des Wortes SEEREISE, erzeugen und anschließend aus dieser Menge mit Hilfe von Cases alle jene Listen auswählen, welche das Muster $\{r, e, i, s, x_5\_, x_6\_, x_7\_, x_8\_\}$ besitzen:

```
Ω = Permutations[{s, e, e, r, e, i, s, e}];
Cases[Ω, {r, e, i, s, x5_, x6_, x7_, x8_}]
Clear[Ω]
```

```
{{r, e, i, s, e, e, s, e}, {r, e, i, s, e, e, e, s},
 {r, e, i, s, e, s, e, e}, {r, e, i, s, s, e, e, e}}
```

4.1.10 Beispiel: Aus einer Urne mit $n = 5$ Kugeln werden nacheinander $k = 4$ Kugeln gezogen, wobei die gezogenen Kugeln nicht zurückgelegt werden. Gesucht ist eine Liste aller jener Realisierungen, bei denen weder beim 2-ten noch beim 4-ten Zug die Kugel mit der Nummer 3 gezogen wird.

▼

Lösung: Das zufällige Ziehen von 4 Kugeln aus einer Urne mit 5 Kugeln, wobei die gezogenen Kugeln nicht zurückgelegt werden, besitzt den Ereignisraum

$$\Omega = \{\{x_1, x_2, x_3, x_4\} \mid x_1, x_2, x_3, x_4 \in \{1, 2, 3, 4, 5\} \text{ und } x_1, x_2, x_3, x_4 \text{ sind paarweise verschieden}\}$$

wobei die Liste $\{x_1, x_2, x_3, x_4\} \in \Omega$ der Realisierung "beim ersten Zug wird die Kugel mit der Nummer x_1 gezogen, beim zweiten Zug wird die Kugel mit der Nummer x_2 gezogen, …, beim vierten Zug wird die Kugel mit der Nummer x_4 gezogen" entspricht. Diese Menge Ω erzeugen wir, indem wir mit Tuples und Range die Menge

$$\Psi = \{\{x_1, x_2, x_3, x_4\} \mid x_1, x_2, x_3, x_4 \in \{1, 2, 3, 4, 5\}\}$$

erzeugen und darauf den Befehl Select mit der Bedingung $\#[[1]] \neq \#[[2]] \neq \#[[3]] \neq \#[[4]]\ \&$ anwenden. Die von uns gesuchte Liste

$$A = \{\{x_1, x_2, x_3, x_4\} \in \Omega \mid x_2 \neq 3 \text{ und } x_4 \neq 3\}$$

erhalten wir dann dadurch, indem wir aus der Liste Ω mit Hilfe von DeleteCases zuerst alle jene Elemente streichen, welche das Muster $\{x_1\_, 3\_, x_3\_, x_4\_\}$ besitzen und anschließend alle jene Elemente streichen, welche das Muster $\{x_1\_, x_2\_, x_3\_, 3\}$ besitzen:

```
Ψ = Tuples[Range[5], 4];
Ω = Select[Ψ, #[[1]] ≠ #[[2]] ≠ #[[3]] ≠ #[[4]] &];
DeleteCases[DeleteCases[Ω, {x1_, 3, x3_, x4_}], {x1_, x2_, x3_, 3}]
Clear[Ψ, Ω]
```

```
{{1, 2, 3, 4}, {1, 2, 3, 5}, {1, 2, 4, 5}, {1, 2, 5, 4}, {1, 4, 2, 5}, {1, 4, 3, 2},
 {1, 4, 3, 5}, {1, 4, 5, 2}, {1, 5, 2, 4}, {1, 5, 3, 2}, {1, 5, 3, 4}, {1, 5, 4, 2},
 {2, 1, 3, 4}, {2, 1, 3, 5}, {2, 1, 4, 5}, {2, 1, 5, 4}, {2, 4, 1, 5}, {2, 4, 3, 1},
 {2, 4, 3, 5}, {2, 4, 5, 1}, {2, 5, 1, 4}, {2, 5, 3, 1}, {2, 5, 3, 4}, {2, 5, 4, 1},
 {3, 1, 2, 4}, {3, 1, 2, 5}, {3, 1, 4, 2}, {3, 1, 4, 5}, {3, 1, 5, 2}, {3, 1, 5, 4},
 {3, 2, 1, 4}, {3, 2, 1, 5}, {3, 2, 4, 1}, {3, 2, 4, 5}, {3, 2, 5, 1}, {3, 2, 5, 4},
 {3, 4, 1, 2}, {3, 4, 1, 5}, {3, 4, 2, 1}, {3, 4, 2, 5}, {3, 4, 5, 1}, {3, 4, 5, 2},
 {3, 5, 1, 2}, {3, 5, 1, 4}, {3, 5, 2, 1}, {3, 5, 2, 4}, {3, 5, 4, 1}, {3, 5, 4, 2},
 {4, 1, 2, 5}, {4, 1, 3, 2}, {4, 1, 3, 5}, {4, 1, 5, 2}, {4, 2, 1, 5}, {4, 2, 3, 1},
 {4, 2, 3, 5}, {4, 2, 5, 1}, {4, 5, 1, 2}, {4, 5, 2, 1}, {4, 5, 3, 1}, {4, 5, 3, 2},
 {5, 1, 2, 4}, {5, 1, 3, 2}, {5, 1, 3, 4}, {5, 1, 4, 2}, {5, 2, 1, 4}, {5, 2, 3, 1},
 {5, 2, 3, 4}, {5, 2, 4, 1}, {5, 4, 1, 2}, {5, 4, 2, 1}, {5, 4, 3, 1}, {5, 4, 3, 2}}
```

4.2 Ermittlung der Wahrscheinlichkeit eines Ereignisses durch Simulation

Wir werden nun an Hand von Beispielen zeigen, wie sich die Simulation einfacher zufälliger Vorgänge zur näherungsweisen Bestimmung der Wahrscheinlichkeit von Ereignissen einsetzen lässt. Üblicherweise besteht diese näherungsweise Bestimmung der Wahrscheinlichkeit eines uns interessierenden Ereignisses durch Simulation in den folgenden drei Schritten:

- Simulation des zu untersuchenden Zufallsexperiments:
Die Ermittlung einer zufälligen Realisierung $\omega \in \Omega$ des zu untersuchenden Zufallsexperiments durch Simulation bereitet in der Regel einige Schwierigkeiten, da man dabei einerseits über das Zufallsexperiment gut Bescheid wissen muss und andererseits in der Lage sein muss, dieses Wissen in eine dem Computer verständliche Befehlskette zu bringen.

- n malige Wiederholung dieses Vorgangs; man erhält dadurch die Stichprobe $\{\omega_1, \omega_2, \ldots, \omega_n\}$:
Mit dem Befehl Table lässt sich der Vorgang "Ermittlung einer zufälligen Realisierung $\omega \in \Omega$ durch Simulation" n mal wiederholen. Der Stichprobenumfang n ist dabei von der gewünschten Genauigkeit abhängig und ist gemäß unserer Faustregel zu wählen. Das Ergebnis ist die Stichprobe $\{\omega_1, \omega_2, \ldots, \omega_n\}$.

- Ermittlung der relativen Häufigkeit des uns interessierenden Ereignisses:
Mit den Befehlen Select, Cases, und DeleteCases sowie passenden Bedingungen bzw Mustern lassen sich aus der so erzeugten Stichprobe $\{\omega_1, \omega_2, \ldots, \omega_n\}$ alle jene Fälle auswählen, welche dem interessierenden Ereignis entsprechen. Die gesuchte relative Häufigkeit lässt sich dann leicht mit Hilfe von Length ermitteln.

An Hand von zahlreichen Beispielen demonstrieren wir nun diesen Vorgang. Da wir die gesuchte Wahrscheinlichkeit stets mit einer Genauigkeit von 10^{-2} ermitteln wollen, sind ge

mäß unserer Faustregel somit jeweils $n = 10^4$ Wiederholungen notwendig. (Zum besseren Verständnis ist es gelegentlich wünschenswert, für einen kleinen Stichprobenumfang n sowohl die durch Simulation gewonnene Stichprobe als auch alle jene Fälle, die dem interessierenden Ereignis entsprechen, aufzulisten. Man entferne dazu nur das Semikolon am Ende unserer Zuweisungen "stichprobe" bzw "ereignis".)

4.2.1 Beispiel: $k = 4$ Würfel werden gleichzeitig geworfen. Wie groß ist die Wahrscheinlichkeit dafür, dass dabei die maximale Augenzahl kleiner als $m = 5$ ist?

▼

Lösung: Analog zu Beispiel 1.2.3 simulieren wir das n-malige Werfen von k Würfeln. Von dieser Stichprobe wählen wir mit Select und der Bedingung Max[#] < m & alle jene Fälle aus, welche dem uns interessierenden Ereignis "die maximale Augenzahl ist kleiner als m" entsprechen:

```
k = 4; m = 5; n = 10000;
stichprobe = Table[RandomInteger[{1, 6}, k], {n}];
ereignis = Select[stichprobe, Max[#] < m &];
Length[ereignis] / n // N
Clear[k, m, n, stichprobe, ereignis]
```

```
0.1945
```

4.2.2 Beispiel: $k = 6$ Würfel werden gleichzeitig geworfen. Wie groß ist die Wahrscheinlichkeit dafür, dass ihre Augensumme größer als $s = 25$ ist?

▼

Lösung: Analog zu Beispiel 1.2.3 simulieren wir das n-malige Werfen von k Würfeln. Von dieser Stichprobe wählen wir mit Select und der Bedingung Apply[Plus, #] > s & alle jene Fälle aus, welche dem uns interessierenden Ereignis "die gewürfelte Augensumme ist größer als s" entsprechen:

```
k = 6; s = 25; n = 10 000;
stichprobe = Table[RandomInteger[{1, 6}, k], {n}];
ereignis = Select[stichprobe, Apply[Plus, #] > s &];
Length[ereignis] / n // N
Clear[k, s, n, stichprobe, ereignis]
```

```
0.1498
```

4.2.3 Beispiel: $k = 15$ Münzen werden gleichzeitig geworfen. Wie groß ist die Wahrscheinlichkeit dafür, dass dabei weniger als $a = 7$ Adler auftreten?

▼

Lösung: Analog zu Beispiel 1.3.3 simulieren wir das n-malige Werfen von k Münzen. Von dieser Stichprobe wählen wir mit Select und der Bedingung Apply[Plus, #] < *a* & alle jene Fälle aus, welche dem uns interessierenden Ereignis "die Anzahl der dabei geworfenen Adler ist kleiner als *a*" entsprechen:

```
k = 15; a = 7; n = 10 000;
stichprobe = Table[RandomInteger[{0, 1}, k], {n}];
ereignis = Select[stichprobe, Apply[Plus, #] < a &];
Length[ereignis] / n // N
Clear[k, a, n, stichprobe, ereignis]
```

```
0.3037
```

4.2.4 Beispiel: Zwei gleich gute Spieler spielen bis zum Gesamtsieg. Dazu muss der erste Spieler noch $a = 8$ und der zweite Spieler noch $b = 6$ Runden gewinnen. Man bestimme die Wahrscheinlichkeit dafür, dass der erste Spieler gewinnt?

▼

Lösung: Wir kommen überein, bei jeder Runde einen Sieg des ersten Spielers mit 1 und einen Sieg des zweiten Spielers mit 0 zu bezeichnen. Da die beiden Spieler gleich gut sind, könnte man das Ergebnis einer Runde ebensogut mit einer Münze auswürfeln. Eine Entscheidung darüber, wer den Gesamtsieg davon trägt, ist nach höchstens $k = a + b - 1$ Runden entschieden. Der erste Spieler gewinnt dabei genau dann, wenn er von diesen k Runden mindestens a Runden gewinnt. Analog zu Beispiel 1.3.3 simulieren wir n mögliche Spielverläufe über k Runden und wählen aus dieser Stichprobe mit Hilfe von Select und der Bedingung Apply[Plus, #] >= *a* & alle jene Spielverläufe aus, welche dem uns interessierenden Ereignis "der erste Spieler trägt den Gesamtsieg davon" entsprechen:

```
a = 8; b = 6; k = a + b − 1; n = 10 000;
stichprobe = Table[RandomInteger[{0, 1}, k], {n}];
ereignis = Select[stichprobe, Apply[Plus, #] ≥ a &];
Length[ereignis] / n // N
Clear[a, b, k, n, stichprobe, ereignis]
```

```
0.2876
```

Für das nächste Beispiel müssen wir klären, was man unter dem im Rahmen der Statistik gelegentlich verwendeten Begriff der Zerlegung einer Liste in ihre Runs versteht:

4.2.5 Definition: Unter der Zerlegung einer Liste in ihre Runs versteht man die Aufspaltung dieser Liste in aufeinanderfolgende Teillisten mit identischen Elementen. Die Zerlegung einer Liste in ihre Runs lässt sich mit dem Befehl Split durchführen.

▼

Beispielsweise gilt

```
Split[{1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0}]
```

```
{{1, 1}, {0}, {1}, {0, 0}, {1}, {0, 0, 0}, {1, 1}, {0}, {1, 1}, {0, 0, 0}}
```

4.2.6 Beispiel: Eine Münze wird $k = 15$ mal geworfen und die Anzahl der auftretenden Runs bestimmt. Wie groß ist die Wahrscheinlichkeit dafür, dass dabei zwischen $r_1 = 5$ und $r_2 = 8$ Runs auftreten?

▼

Lösung: Analog zu Beispiel 1.3.3 simulieren wir das n-malige Werfen von k Münzen. Von dieser Stichprobe wählen wir mit Select und der Bedingung r1 ≤ Length[Split[#]] ≤ r2 & alle jene Fälle aus, die dem uns interessierenden Ereignis "die Anzahl r der Runs genügt der Bedingung $r_1 \leq r \leq r_2$" entsprechen:

```
k = 15; r1 = 5; r2 = 8; n = 10 000;
stichprobe = Table[RandomInteger[{0, 1}, k], {n}];
ereignis = Select[stichprobe, r1 ≤ Length[Split[#]] ≤ r2 &];
Length[ereignis] / n // N
Clear[k, r1, r2, n, stichprobe, ereignis]
```

```
0.5741
```

4.2.7 Beispiel: Aus einer Urne mit $m = 8$ Kugeln werden $k = 6$ Kugeln gezogen, wobei die gezogene Kugel stets wieder in die Urne zurückgelegt wird. Wie groß ist die Wahrscheinlichkeit dafür, dass dabei mehr als $v = 4$ verschiedene Kugeln gezogen werden?

▼

Lösung: Analog zu Beispiel 1.4.1 simulieren wir das Ziehen mit Zurücklegen von k Kugeln aus einer Urne mit m Kugeln und wiederholen diesen Vorgang unter Verwendung von Table n mal. Aus dieser Stichprobe wählen wir mit Select und der Bedingung Length[Union[#]] > v & alle jene Fälle aus, die dem uns interessierenden Ereignis "es werden mehr als v verschiedene Kugeln gezogen" entsprechen (wendet man den Befehl Union auf eine Liste an, so werden alle mehrfach vorkommenden Elemente dieser Liste gestrichen und die verbleibenden Elemente in natürlicher Reihenfolge angeordnet):

```
k = 6; m = 8; v = 4; n = 10 000;
stichprobe = Table[RandomInteger[{1, m}, k], {n}];
ereignis = Select[stichprobe, Length[Union[#]] > v &];
Length[ereignis] / n // N
Clear[k, m, v, n, stichprobe, ereignis]
```

```
0.4656
```

4.2.8 Beispiel: Aus einer Urne mit $m = 10$ Kugeln werden $k = 5$ Kugeln gezogen, wobei die gezogene

Kugel stets wieder in die Urne zurückgelegt wird. Wie groß ist die Wahrscheinlichkeit dafür, dass dabei mindestens eine Kugel mehrfach gezogen wird?

▼

Lösung: Analog zu Beispiel 1.4.1 simulieren wir das Ziehen mit Zurücklegen von k Kugeln aus einer Urne mit m Kugeln und wiederholen diesen Vorgang unter Verwendung von Table n mal. Aus dieser Stichprobe wählen wir mit Select und der Bedingung Length[Union[#]] < k & alle jene Fälle aus, die dem uns interessierenden Ereignis "es wird mindestens eine Kugel mehrfach gezogen" entsprechen:

```
k = 5; m = 10; n = 10 000;
stichprobe = Table[RandomInteger[{1, m}, k], {n}];
ereignis = Select[stichprobe, Length[Union[#]] < k &];
Length[ereignis] / n // N
Clear[k, m, n, stichprobe, ereignis]
```

```
0.6986
```

4.2.9 Beispiel: Wie groß ist die Wahrscheinlichkeit dafür, dass in einer Gesellschaft von $k = 50$ Personen mindestens zwei Personen am gleichen Tag Geburtstag haben? Dabei nehmen wir der Einfachheit halber an, dass die Geburten gleichmäßig über das Jahr verteilt sind und vernachlässigen Zwillingsgeburten.

▼

Lösung: Die Frage "Wie groß ist die Wahrscheinlichkeit dafür, dass in einer Gesellschaft von $k = 50$ Personen mindestens zwei Personen am gleichen Tag Geburtstag haben?" entspricht der Frage "Wie groß ist die Wahrscheinlichkeit dafür, dass beim Ziehen mit Zurücklegen von $k = 50$ Kugeln (Geburtstage der Personen dieser Gesellschaft) aus einer Urne mit $m = 365$ Kugeln (Tage des Jahres) mindestens eine Kugel (Geburtstag) mehrfach gezogen wird?" Diese Fragestellung wurde bereits in Aufgabe 4.2.8 behandelt:

```
k = 50; m = 365; n = 10 000;
stichprobe = Table[RandomInteger[{1, m}, k], {n}];
ereignis = Select[stichprobe, Length[Union[#]] < k &];
Length[ereignis] / n // N
Clear[k, m, n, stichprobe, ereignis]
```

```
0.9694
```

4.2.10 Beispiel: Auf $m = 5$ anfangs leere Urnen werden $k = 8$ Kugeln verteilt, wobei jede Kugel mit der gleichen Wahrscheinlichkeit in jede Urne gelangen kann. Wie groß ist die Wahrscheinlichkeit dafür, dass dabei genau $z = 2$ Urnen leer bleiben?

▼

Lösung: Jede Realisierung unseres Zufallsexperiments lässt sich durch eine Liste der Form $\{x_1, x_2, \ldots, x_k\}$ mit $x_i \in \{1, 2, \ldots, m\}$ beschreiben, wobei diese Liste der Realisierung "die erste Kugel gelangt in die x_1-te Urne, …, die k-te Kugel gelangt in die x_k-te Urne" entspricht und wobei alle derartigen Realisierungen gleichwahrscheinlich sind. Offenbar bleiben dabei genau dann $z = 2$ Urnen leer, wenn die Menge $\{x_1, x_2, \ldots, x_k\}$ die Mächtigkeit $v = m - z = 3$ besitzt. Die gesuchte Wahrscheinlichkeit ermitteln wir analog zu Beispiel 4.2.7:

```
k = 8; m = 5; v = 3; n = 10 000;
stichprobe = Table[RandomInteger[{1, m}, k], {n}];
ereignis = Select[stichprobe, Length[Union[#]] == v &];
Length[ereignis] / n // N
Clear[k, m, v, n, stichprobe, ereignis]
```

```
0.1475
```

4.2.11 Beispiel: Einer Urne mit $r = 5$ roten und $s = 15$ schwarzen Kugeln werden gleichzeitig $k = 7$ Kugeln entnommen. Wie groß ist die Wahrscheinlichkeit dafür, dass dabei mindestens eine rote Kugel gezogen wird?

▼

Lösung: Wir nummerieren die $m = r + s$ Kugeln in der Weise, dass die r roten Kugeln die Nummern 1, 2, …, r und die s schwarzen Kugeln die Nummern $r + 1$, $r + 2$, …, m erhalten. Analog zu Beispiel 1.4.2 lässt sich das Ziehen ohne Zurücklegen von k Kugeln aus dieser Urne mit $m = r + s$ Kugeln simulieren. Mit Hilfe von Table wiederholen wir diesen Vorgang n mal. Aus dieser Stichprobe wählen wir mit Hilfe von Select alle jene Realisierungen aus, welche mit der Menge {1, 2, …, r} einen nicht leeren Durchschnitt besitzen, welche also der Bedingung Length[♯ ⋂ Range[r]] ≥ 1 & genügen:

```
k = 7; r = 5; s = 15; m = r + s; n = 10 000;
stichprobe = Table[RandomSample[Range[m], k], {n}];
ereignis = Select[stichprobe, Length[♯ ⋂ Range[r]] ≥ 1 &];
Length[ereignis] / n // N
Clear[k, r, s, m, n, stichprobe, ereignis]
```

```
0.9216
```

4.2.12 Beispiel: In einer Urne befinden sich $m = 45$ mit den Zahlen 1, 2, …, m nummerierte Lose. Aus dieser Urne werden ohne Zurücklegen $k = 6$ Lose gezogen. Wie groß ist die Wahrscheinlichkeit dafür, dass die größte dabei gezogene Zahl kleiner als $g = 25$ ist?

▼

Lösung: Analog zu Beispiel 4.2.11 simulieren wir das n-malige Ziehen ohne Zurücklegen von k Kugeln aus einer Urne mit m Kugeln. Aus dieser Stichprobe wählen wir mit Select und der Bedingung Max[♯] < g & alle jene Fälle aus, die dem uns interessierenden Ereignis "die größte dabei gezogene Zahl ist kleiner als g" entsprechen:

```
k = 6; m = 45; g = 25; n = 10 000;
stichprobe = Table[RandomSample[Range[m], k], {n}];
ereignis = Select[stichprobe, Max[♯] < g &];
Length[ereignis] / n // N
Clear[k, m, g, n, stichprobe, ereignis]
```

```
0.0177
```

4.2.13 Beispiel: Eine Lotterie umfasst $m = 50$ Lose, von denen $g = 5$ gewinnen. Wie groß ist die Wahrscheinlichkeit für (mindestens) einen Gewinn, wenn man $k = 4$ Lose kauft?

▼

Lösung: Wir nummerieren die $m = 50$ Lose mit den Zahlen 1, 2, …, 50, wobei wir den $g = 5$ Gewinnlosen die Zahlen 1, 2, …, g zuordnen. Analog zu Beispiel 4.2.11 lässt sich das n-malige zufällige Auswählen von $k = 4$ Losen aus $m = 50$ Losen simulieren. Aus dieser Stichprobe wählen wir mit Select unter Verwendung der Bedingung Min[#] ≤ *g* & alle jene Fälle aus, die dem uns interessierenden Ereignis "es wird (mindestens) ein Gewinnlos gezogen" entsprechen:

```
k = 4; m = 50; g = 5; n = 10 000;
stichprobe = Table[RandomSample[Range[m], k], {n}];
ereignis = Select[stichprobe, Min[#] ≤ g &];
N[Length[ereignis] / n]
Clear[k, m, g, n, stichprobe, ereignis]
```

```
0.3502
```

4.2.14 Beispiel: 12 Damen und 6 Herren werden in drei Gruppen zu je sechs Personen aufgeteilt. Wie groß ist die Wahrscheinlichkeit dafür, dass dabei in jede dieser Gruppen genau zwei Herren gelangen?

▼

Lösung: Wir nummerieren die 18 zu vergebenden Plätze mit den Zahlen 1, 2, …, 18, wobei der ersten Gruppe die Plätze 1, 2, …, 6, der zweiten Gruppe die Plätze 7, 8, …, 12 und der dritten Gruppe die restlichen Plätze 13, 14, …, 18 zugeordnet werden. Unser Zufallsexperiment besteht im zufälligen Aufteilen der 6 Herren auf diese 18 Plätze, wofür

$$\Omega = \{\{x_1, x_2, \ldots, x_6\} \mid x_1, x_2, \ldots, x_6 \in \{1, 2, \ldots, 18\} \text{ und } x_1 < x_2 < \ldots < x_6\}$$

ein passender Ereignisraum ist. Die Liste $\{x_1, x_2, \ldots, x_6\} \in \Omega$ entspricht dabei der Realisierung "den 6 Herren werden die Plätze $x_1, x_2, \ldots, x_6$ zugewiesen". Das uns interessierende Ereignis "in jeder dieser drei Gruppen befinden sich genau zwei Herren" entspricht damit der Menge

$$A = \{\{x_1, x_2, \ldots, x_6\} \in \Omega \mid x_1, x_2 \leq 6 \,\wedge\, 7 \leq x_3, x_4 \leq 12 \,\wedge\, 13 \leq x_5, x_6\}$$

Analog zu Beispiel 1.4.2 simulieren wir zuerst das Ziehen ohne Zurücklegen von $k = 6$ Kugeln (Herren) aus einer Urne mit $m = 18$ Kugeln (Plätze) und wenden anschließend auf jede dieser Realisierungen den Befehl Sort an. Mit Hilfe von Table wiederholen wir diesen Vorgang n mal. Aus dieser Stichprobe wählen wir mit Select und der Bedingung #[[2]] ≤ 6 < #[[3]] ⋀ #[[4]] ≤ 12 < #[[5]] & alle jene Fälle aus, die dem uns interessierenden Ereignis "in jede der drei Gruppen gelangen genau zwei Herren" entsprechen:

```
k = 6; m = 18; n = 10 000;
stichprobe = Table[Sort[RandomSample[Range[m], k]], {n}];
ereignis = Select[stichprobe, #[[2]] ≤ 6 < #[[3]] ⋀ #[[4]] ≤ 12 < #[[5]] &];
Length[ereignis] / n // N
Clear[k, m, n, stichprobe, ereignis]
```

```
0.1849
```

4.2.15 Beispiel: $m = 8$ Ehepaare besuchen eine Party. Für ein Tanzspiel werden die Tanzpartner zufällig ausgelost, wodurch jeder Herr jede Dame mit der gleichen Wahrscheinlichkeit zugeteilt bekommt. Wie groß ist die Wahrscheinlichkeit dafür, dass kein Herr mit seiner eigenen Frau tanzt?

▼

Lösung: Wir nummerieren die $m = 8$ Ehepaare mit den Zahlen 1, 2, …, m. Die Menge der möglichen Kombinationen von Tanzpartnern lässt sich dann durch die Menge

$$\Omega = \{\{x_1, x_2, \ldots, x_m\} \mid x_i \in \{1, 2, \ldots, m\} \text{ und } x_1, x_2, \ldots, x_m \text{ sind paarweise verschieden}\}$$

beschreiben, wobei die Liste $\{x_1, x_2, \ldots, x_m\} \in \Omega$ der Realisierung "der erste Herr tanzt mit der x_1-ten Dame, der zweite Herr tanzt mit der x_2-ten Dame, …, der m-te Herr tanzt mit der x_m-ten Dame" entspricht. Diese zufällige Bildung von Tanzpaaren entspricht offenbar einer zufälligen Permutation von m Dingen (Damen) und lässt sich analog zu Beispiel 1.4.2 simulieren. Mit Table wiederholen wir dieses zufällige Auslosen der Tanzpaare n mal und wählen aus dieser Stichprobe mit Select und der Bedingung Apply[And, Table[#[[i]] ≠ i, {i, m}]] & alle jene Fälle aus, die dem uns interessierenden Ereignis "kein Herr tanzt mit seiner eigenen Frau" entsprechen:

```
m = 8; n = 10 000;
stichprobe = Table[RandomSample[Range[m]], {n}];
ereignis = Select[stichprobe, Apply[And, Table[#[[i]] ≠ i, {i, m}]] &];
Length[ereignis] / n // N
Clear[m, n, stichprobe, ereignis]
```

```
0.359
```

4.2.16 Beispiel: Ein Teilchen bewegt sich mit der Geschwindigkeit 1 auf der Zahlengerade. Kommt es in einen Punkt mit ganzzahliger Abszisse, so entscheidet sich unser Teilchen (unabhängig von seinem bisherigen Weg) mit gleicher Wahrscheinlichkeit für ein Weiterwandern nach rechts bzw links. Das Teilchen beginnt mit seiner Wanderung im Ursprung. Wie groß ist die Wahrscheinlichkeit dafür, dass sich das Teilchen zum Zeitpunkt $k = 15$ rechts vom Punkt $r = 2$ befindet?

▼

Lösung: Wir kommen überein, ein Weiterwandern unseres Teilchens nach rechts mit +1 und nach links mit −1 zu bezeichnen. Jede Entscheidung des Teilchens für ein Weiterwandern stellt damit eine auf der Menge $\{+1, -1\}$ gleichverteilte Zufallszahl dar. Mit Hilfe von RandomInteger und Table simulieren wir n zufällige Wanderungen unseres Teilchens. Von dieser Stichprobe wählen wir mit Select und der Bedingung Apply[Plus, #] > r & alle jene Fälle aus, in denen die Wanderung unseres Teilchens rechts vom Punkt r endet:

```
k = 15; r = 2; n = 10 000;
stichprobe = Table[2 RandomInteger[{0, 1}, k] − 1, {n}];
ereignis = Select[stichprobe, Apply[Plus, #] > r &];
Length[ereignis] / n // N
Clear[k, n, r, stichprobe, ereignis]
```

```
0.2892
```

Gelegentlich benötigt man beim Simulieren Abfragen zur Flusskontrolle. Derartige Abfragen lassen sich in *Mathematica* mit den Befehlen For und Which durchführen:

■ For[*start*, *test*, *expr*]

führt zuerst *start* aus und evaluiert anschließend solange *expr*, bis der Test *test* den Wahrheitswert falsch liefert.

■ Which[$test_1$, $value_1$, $test_2$, $value_2$, …]

evaluiert nacheinander die Tests $test_i$ und gibt jenen Wert $value_i$ aus, der zum ersten Test gehört, welcher den Wahrheitswert wahr liefert.

4.2.17 Beispiel: Mit einer Münze wird so lange geworfen, bis erstmals hintereinander drei Adler auftreten.

Wie groß ist die Wahrscheinlichkeit dafür, dass dazu mehr als $w = 15$ Würfe notwendig sind?

▼

Lösung: Will man dieses Zufallsexperiment simulieren, so muss man mit Hilfe von RandomInteger so lange auf der Menge {0, 1} gleichverteilte Zufallszahlen erzeugen, bis erstmals die drei zuletzt erzeugten Zufallszahlen gleich 1 sind. Unter Verwendung von Module und For schreiben wir zuerst ein kleines Unterprogramm, mit dem sich eine derartige Wurfserie simulieren lässt. Dabei bedeutet

- *start* die Erzeugung von drei auf der Menge {0, 1} gleichverteilten Zufallszahlen mittels RandomInteger;
- *test* die Überprüfung, ob die drei zuletzt erzeugten Zufallszahlen gleich {1, 1, 1} sind;
- *expr* das Ergänzen der bisher erzeugten Zufallszahlen durch eine weitere Zufallszahl mit Hilfe von Append.

```
wurfserie := Module[{w}, For[
      w = RandomInteger[{0, 1}, 3],
      {w[[-1]], w[[-2]], w[[-3]]} =!= {1, 1, 1},
      w = Append[w, RandomInteger[]]]; w]

wurfserie
```

```
{1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1}
```

Mit Hilfe von Table simulieren wir nun n derartige Wurfserien. Aus der auf diese Weise erzeugten Stichprobe wählen wir mit Select und der Bedingung Length[#] > w & alle jene Fälle aus, die dem uns interessierenden Ereignis "es sind mehr als w Würfe notwendig, bis erstmals hintereinander drei Adler auftreten" entsprechen:

```
w = 15; n = 10 000;
stichprobe = Table[wurfserie, {n}];
ereignis = Select[stichprobe, Length[#] > w &];
Length[ereignis] / n // N
Clear[w, n, stichprobe, ereignis]
```

```
0.3221
```

```
Clear[wurfserie]
```

4.2.18 Beispiel: Aus einer Urne mit $s = 8$ schwarzen, $r = 6$ roten und $g = 4$ grünen Kugeln werden so lange Kugeln gezogen und nach jedem Zug wieder zurückgelegt, bis erstmals hintereinander zwei grüne Kugeln gezogen werden. Wie groß ist die Wahrscheinlichkeit dafür, dass dazu höchstens $h = 30$ Züge notwendig sind.

▼

Lösung: Wir nummerieren die $s = 8$ schwarzen Kugeln mit den Zahlen 1, 2, …, 8, die $r = 6$ roten Kugeln mit den Zahlen 9, 10, …, 14 und die $g = 4$ grünen Kugeln mit den Zahlen 15, 16, …, 18. Mit Hilfe von RandomInteger und Which erzeugen wir zuerst eine Funktion, mit der sich die Farbe der gezogenen Kugel ermitteln lässt:

```
farbe[s_, r_, g_] := Which[x = RandomInteger[{1, s + r + g}]; x ≤ s, schwarz, x ≤ s + r, rot, x > s + r, grün]

farbe[8, 6, 4]
```

```
rot
```

Die eigentliche Simulation erfolgt dann ähnlich wie in Beispiel 4.2.17, indem wir zuerst unter Verwendung von Module und For ein kleines Unterprogramm schreiben, mit dem sich eine derartige Zugserie simulieren lässt

```
zugserie[s_, r_, g_] := Module[{zug}, For[
    zug = Table[farbe[s, r, g], {2}], {zug[[-1]], zug[[-2]]} =!= {grün, grün}, zug = Append[zug, farbe[s, r, g]]]];

zugserie[8, 6, 4]
```

```
{schwarz, schwarz, grün, schwarz, schwarz, rot, schwarz, rot, grün, grün}
```

anschließend mit Hilfe von Table *n* derartige Zugserien simulieren und aus der auf diese Weise erzeugten Sichprobe mittels Select und der Bedingung Length[#] ≤ *h* & alle jene Fälle auswählen, die dem uns interessierenden Ereignis "es sind höchstens *h* Würfe notwendig, bis erstmals zweimal hintereinander eine grüne Kugel gezogen wird" entsprechen:

```
s = 8; r = 6; g = 4; h = 30; n = 10 000;
stichprobe = Table[zugserie[s, r, g], {n}];
ereignis = Select[stichprobe, Length[#] ≤ h &];
Length[ereignis] / n // N
Clear[s, r, g, h, n, stichprobe, ereignis]
```

```
0.7056
```

```
Clear[farbe, zugserie]
```

In manchen Fällen leistet der Befehl FoldList gute Dienste:

■ FoldList[f, x, {y_1, y_2, …}]

erzeugt die Liste $\{x,\ f[x, y_1],\ f[f[x, y_1], y_2], \dots\}$.

▼

Beispielsweise gilt

```
FoldList[Plus, 1, {1, 2, 3, 4, 5, 6, 7}]
```

```
{1, 2, 4, 7, 11, 16, 22, 29}
```

4.2.19 Beispiel: Vor der Kasse eines Theaters stehen $m = a + b$ Personen, von denen $a = 8$ nur einen 10 € Schein und $b = 12$ nur einen 20 € Schein bei sich haben. Eine Karte kostet 10 €. In der Kasse sind zu Beginn des Kartenverkaufs $u = 5$ 10 € Scheine. Wie groß ist die Wahrscheinlichkeit dafür, dass der Kartenverkauf ohne Stockung abläuft, also in einem Moment, in dem sich in der Kasse kein 10 € Schein befindet, keine Person eine Karte kaufen will, die nur einen 20 € Schein bei sich hat?

▼

Lösung: Wir kommen überein, eine Person mit einem 10 € Schein mit 1 und eine Person mit einem 20 € Schein mit −1 zu bezeichnen. Unter Verwendung von Module schreiben wir zuerst ein kleines Unterprogramm, mit dem sich die Anzahl der während des Kartenverkaufs jeweils in der Kasse vorhandenen 10 € Scheine simulieren lässt. Dazu erzeugen wir mit den Befehlen Join und Table eine Liste der Form

$$\{1, \underbrace{1, \dots}_{a\text{ mal}}, 1, -1, \underbrace{-1, \dots}_{b\text{ mal}}, -1\}$$

wenden auf diese Liste den Befehl RandomSample an (diese Liste entspricht einer zufälligen Anordnung der m Personen vor der Theaterkasse) und wenden darauf schließlich den Befehl FoldList mit dem Startwert $u = 5$ und der Funktion Plus an:

```
kartenverkauf[a_, b_, u_] := Module[{liste},
   liste = RandomSample[Join[Table[1, {a}], Table[−1, {b}]]];
   FoldList[Plus, u, liste]]

kartenverkauf[8, 12, 5]
```

```
{5, 6, 5, 4, 5, 6, 5, 6, 5, 6, 7, 6, 7, 6, 7, 6, 5, 4, 3, 2, 1}
```

Wir erzeugen nun mit Hilfe von Table n derartige Listen und wählen aus dieser Stichprobe mit Select und der Bedingung Min[#] ≥ 0 & alle jene Fälle aus, in denen der Kartenverkauf ohne Stockung abläuft:

```
a = 8; b = 12; u = 5; n = 10 000;
stichprobe = Table[kartenverkauf[a, b, u], {n}];
ereignis = Select[stichprobe, Min[#] ≥ 0 &];
Length[ereignis] / n // N
Clear[a, b, u, n, stichprobe, ereignis]
```

```
0.6911
```

```
Clear[kartenverkauf]
```