

APL-Journal

A Programming Language

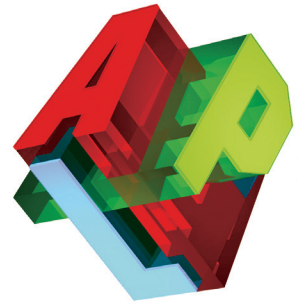
1/2 2009

Herbert Voß

The current state of the PStricks project

Markos Mitsos

Analyse von Kniffel und Ermittlung der optimalen
Spielstrategie über APL



Im Blickpunkt

IBM

Workstation APL2 Version 2 Service Level 17

APL-Journal

28. Jg. 2009, ISSN 1438-4531

Herausgeber: Rainer Nußbaum,
APL-Germany e.V., Mannheim,
<http://www.apl-germany.de>

Redaktion: Dipl.-Volksw. Martin
Barghoorn (verantw.), TU Berlin,
Franklinstr. 28, D-10587 Berlin,
Tel. (030) 314 24392, Fax (030)
314 25901

Verlag: RHOMBOS-VERLAG, Ber-
lin, Kurfürstenstr. 17, D-10785 Ber-
lin, Tel. (030) 261 9461, Fax (030)
261 6300, eMail: verlag@rhombos.de,
Internet: www.rhombos.de

Erscheinungsweise: halbjährlich

Erscheinungsort: Berlin

Gestaltung: RHOMBOS-VERLAG

Druck: dbusiness.de GmbH, Berlin

Copyright: APL Germany e.V. (für
alle Beiträge, die als Erstveröffent-
lichung erscheinen)

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutzgesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürfen. Eine Haftung für die Richtigkeit der veröffentlichten Informationen kann trotz sorgfältiger Prüfung von Herausgeber und Verlag nicht übernommen werden.

Mit Namen gekennzeichnete Artikel geben nicht unbedingt die Meinung des Herausgebers oder der Redaktion wieder. Für unverlangte Einsendungen wird keine Haftung übernommen. Nachdruck ist nur mit Zustimmung des Herausgebers sowie mit Quellenangabe und Einsendung eines Beleges gestattet. Überarbeitungen eingesandter Manuskripte liegen im Ermessen der Redaktion.

INHALT

Herbert Voß

The current state of the PSTricks project

PSTricks is a shortcut for PostScript Tricks and uses the enormous graphical skills of the old programming language PostScript. It is a so-called page code language (PCL) which is distributed since 1984 by Adobe Systems. PostScript is fully Turing compatible and a stack orientated programming language, just like Assembler. It works in the same way as the Hewlett Packard pocket calculators work. PDF is derived from PostScript with some more important extensions but without the possibility of mathematical calculations.

Markos Mitsos

Analyse von Kniffel und Ermittlung der optimalen Spielstrategie über APL

Spielstrategie über APL

Das Spiel Kniffel ist zwar ein Glücksspiel, es enthält jedoch eine Entscheidung, welche der Spieler selbst treffen kann: Die Anzahl der Würfel, welche bei einem Zug zum Nachwurf verwendet werden. In diesem Artikel wird erklärt, wie dieses Spielelement verwendet werden kann, um, bedingte Wahrscheinlichkeiten nutzend, eine optimale Spielstrategie zu entwickeln.

Zu Beginn werden einige einfache Überlegungen zum Erstwurf angestellt, und gezeigt wie die Argumentation der elementaren Wahrscheinlichkeitslehre durch genauso elementare APL-Algorithmen ersetzt werden kann. Der Vorteil letzterer ist, dass sie auch beliebig komplexe Spielstellungen bewerten können. Eine solche Bewertung kann so viele Fallunterscheidungen benötigen, dass sie praktisch nicht mit Papier und Kugelschreiber zu bewältigen ist.

Neues in Workstation APL2 V2.0

2

The current state of the PSTricks project

3

Analyse von Kniffel und Ermittlung der optimalen Spielstrategie über APL

17

Impressum

52

Bildnachweis:

Stephanie-Hofschlaeger_pixelio.de (S. 46), Ingrid-Ruthe_pixelio.de (S. 55),

Workstation APL2 Version 2 Service Level 17

The IBM APL Products and Services group is pleased to announce Service Level 17 for Workstation APL2 for Multiplatforms Version 2.0. This new service level is available for all Workstation APL2 platforms including AIX, Linux, Sun Solaris, and Windows.

Along with fixes for customer reported problems, Service Level 17 contains the following enhancements:

All platforms:

- * The performance of the Find primitive is improved for right arguments of rank 2 or higher.
- * When an APL error occurs while running under the Calls to APL2 facility, the execution stack is now available to the calling program. When calling APL2 from APL2, the stack is always returned in the error result. >From C, use the EXECUTE_APL_ESTACK callback service to enable this facility. >From Java, examine the Stack field of the APL2exception class. >From Visual Basic, use the GetStack function.
- * A new tool for printing APL objects has been added to the DEMOJAVA workspace.
- * The XML2APL function in DEMOJAVA now supports DTD addresses.

Windows:

- * A session manager system option has been added to control the workspace size of watch windows.
- * The line number popup window in the object editor has new choices for Trace this line and Stop this line.
- * New key shortcuts include Ctrl+Double Click to mark a token, Ctrl+Shift+Double Click to localize or unlocalize a name, and Alt+PageDown to move the current line to the top of the window.

Further information on the new facilities will be found in the updated APL2 User's Guide and on-line help after installing Service Level 17. A complete list of fixes included in this level will be found in the updated README (Unix systems) or Service Information (Windows systems) file.

The APL2 Service Level 17 download is available to customers with a Software Maintenance contract, through the Support link at the APL2 web page, <http://www.ibm.com/software/awdtools/apl>

AIX, APL2 and IBM are trademarks of the IBM Corporation.

Linux is a trademark of Linus Torvalds.

Java, Sun and Solaris are trademarks of Sun Microsystems, Inc.

Visual Basic and Windows are trademarks of the Microsoft Corporation.

Herbert Voß

The current state of the PSTricks project

PSTricks is a shortcut for PostScript Tricks and uses the enormous graphical skills of the old programming language PostScript. It is a so-called page code language (PCL) which is distributed since 1984 by Adobe Systems. PostScript is fully Turing compatible and a stack orientated programming language, just like Assembler. It works in the same way as the Hewlett Packard pocket calculators work. PDF is derived from PostScript with some more important extensions but without the possibility of mathematical calculations.

1 Creating a graphic with PSTricks

TeX as a typesetting machine cannot make a full use of the possibilities of PostScript. There cannot be a direct interaction between TeX and PostScript, it is more a one way communication from TeX to PostScript. The interface between these two systems is the the DVI converter dvips which converts the dvi output of TeX into the PostScript format. The DVI format has only a temporarily state, it is not really needed and only of some historical interest. On TeX side the user has to reserve some space (a box) which is filled on PostScript side with a graphic or some text. If this space is not reserved then everything will be printed over the text, depending on the current point before the PostScript related code. This PostScript code must be transferred to from TeX via the DVI output with the macro `\special`. Its contents is ignored by TeX and passed to PostScript where it will be executed by a PostScript interpreter like Ghostscript or Distiller.

Figure 2 on page 1014 shows the important way of the TeX source to the destination format PDF. When using a graphical user interface for editing the TeX source one can choose the output format PDF, but has to pay attention that it is the way `LaTeX-dvips-ps2pdf`. For three often used graphical user interfaces (GUI) the Figure 1 on page 1001 shows the preferences for this way of compiling a LaTeX source document.

The first example shows the use of PostScript related specials without reserving any space on TeX level. In the beginning and in the end there is a named node which can be connect by a line or a curve, in this by a curve with an arrow and stroke opacity. It is drawn over the text, there was no space on TeX level reserved. In examples like these it may

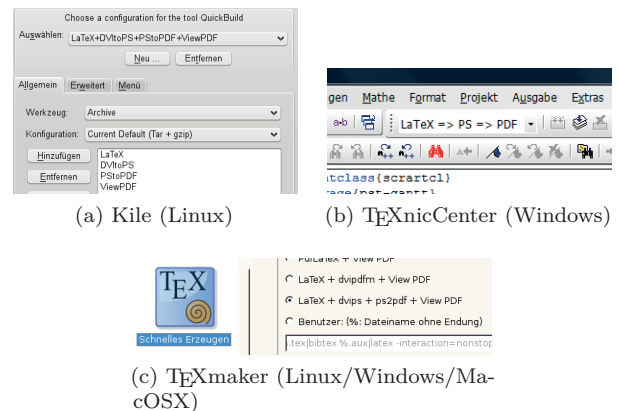


Figure 1: Preferences for the destination format PDF via the intermediate format DVI.

be useful to draw directly over the main text part of the document.

The beginning of all: Hello, here is some text without a meaning. This text should show, how a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like »Huardest gefburn«. Kjift – Never mind! A blind text like this gives you information about the selected font, how the letters are written and the impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for a special contents, but the length of words should match to the language. and this is the end my friend.

```
\usepackage{blindtext,pst-node}
\SpecialCoor

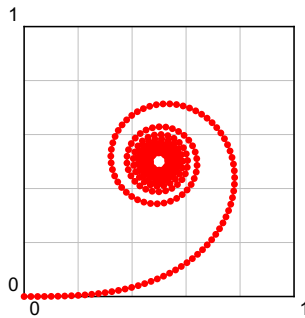
\node{A}{The beginning of all: }
\blindtext \node{E}{ and this is
the end my friend.}
\ncurve[linewidth=5pt,
strokeopacity=0.4,ncurvB=0.9,
arrowscale=1.5,arrows=-D>]{A}{E}
```

The advantage of PSTricks in comparison to METAPOST or tikZ is the possibility to use all features of a powerful programming language with a very good support for graphic operations. This is the reason why *any* calculation with mathematical functions or large external data sets can be done before the output is printed. Solving a differential equation on the fly is also possible as drawing three dimensional solids with hidden lines and surfaces; everything is done on PostScript side. In the TeX or LaTeX source one has only to define the space of the box and to describe the code with TeX or LaTeX macros which are then passed as specials to

PostScript. The next example shows the output of the solution of the differential equation system of first order:

$$\dot{x} = \cos \frac{\pi}{2} \cdot x^2 \quad (1)$$

$$\dot{y} = \sin \frac{\pi}{2} \cdot x^2 \quad (2)$$



```
\usepackage{pstricks-add}
\psset{unit=5}
\begin{pspicture}(-0.04,-0.04)(1,1)
\psgrid[subgriddiv=5,subgridcolor=lightgray]
\psplotDiffEqn[whichabs=0,whichord=1,linewidth=red,
method=rk4,algebraic,plotpoints=400,
showpoints=true]{0}{10}{0 0}%
{cos(Pi*x^2/2)|sin(Pi*x^2/2)}
\end{pspicture}
```

The parameters (coordinates) of the Environment `pspicture` have a different meaning for \TeX and PostScript; for \TeX they define width and height or the box which has its lower left side at the current point in \TeX . When there is no shift defined, then the lower side of this box is always on the baseline of the current text line. It is for \TeX just the same as a box for a single letter. \TeX needs the coordinates only for the formatting, what will be inserted later into this box is not of interest to \TeX .

For PostScript the coordinates define a two dimensional area with the lower left and upper right corner of the rectangle. The origin of this cartesian coordinate system may be inside or outside of this rectangle; it depends on the values of the coordinates. An example: `begin{pspicture}(-1,-2)(4,4)` defines for \TeX a box with a width of five length units ($4 - (-1)$) and a height of six length units ($4 - (-2)$). For PostScript the origin of this box is one length unit to the right and two length units up, measured from the current point, which is the lower left of the \TeX box.

The next example defines on \TeX level a box with a width and height of $2,5\text{cm} \times 1,5\text{cm}$.

The box with a reserved space of $2,5\text{cm} \times 1,5\text{cm}$ is by definition with its lower left side at the current point. The lower side is on the baseline, which can easily be



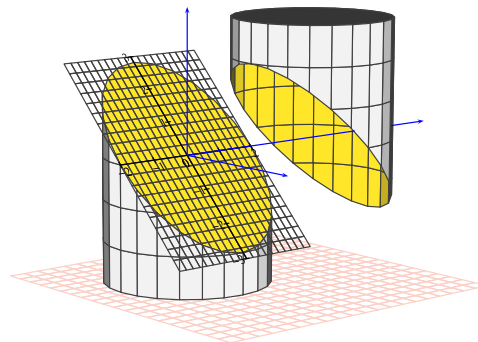
seen on the this box: its internal origin maybe somewhere, also outside this box. In this example the origin is at $(1,0.5)$ units, measured from the lower left corner of the box.

```
\usepackage{pstricks,pst-plot}
\raggedright The box with a reserved space
of  $2,5\text{cm} \times 1,5\text{cm}$  is by definition with
its lower left side at the current point.
The lower side is on the baseline, which
can easily be seen on the this box:
\psframebox[framesep=0]{%
\begin{pspicture}(-1,-0.5)(1.5,1)
\psaxes[labels=none]{->}(0,0)(-1,-0.5)(1.5,1)
\end{pspicture}}
its internal origin mybe somewhere, also
outside this box. In this example the origin
is at \texttt{(1,0.5)} units, measured from
the lower left corner of the box.
```

There are several optional arguments for the main environment `pspicture`:

```
\pspicture * [Options] (xMin,yMin) (xMax,yMax)
...
\end{pspicture}
\pspicture * [Options] (xMax,yMax)
...
\end{pspicture}
\begin{pspicture * } [Options] (xMin,yMin) (xMax,yMax)
...
\end{pspicture * }
\begin{pspicture * } [Options] (xMax,yMax)
...
\end{pspicture * }
```

Another main focus of `PSTricks` are three dimensional graphic objects with supporting the hidden line algorithm for lines and surfaces, too. The powerful package `pst-solides3d` allows to combine any three dimensional solid, given by coordinates or a mathematical expression. The following example shows the book cover image from [2], a cylindrical object is divided by a plane into two parts. One of it is moved and rotated.




```
\usepackage[dvipsnames]{pstricks}
\usepackage{pst-solides3d}

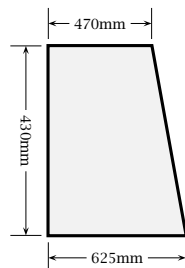
\begin{pspicture}[solidmemory](-4,-5)(7,4)
\psset{viewpoint=50 -40 10 rtp2xyz,Decran=50,
  linecolor=darkgray,lightsrc=viewpoint}
\psSolid[object=grille,action=draw,base=-3 5 -3 5,
  linecolor=Salmon!40](0,0,-3)
\psSolid[object=cylindre,r=2,h=6,ngrid=6 24,
  plansepare={[0.707 0 0.707 0]},name=Zylinder,
  action=none](0,0,-3)
\psSolid[object=load,load=Zylinder1,
  fillcolor=black!5,fc0l=0 (Goldenrod)]
\psSolid[object=load,load=Zylinder0,RotZ=90,
  fillcolor=black!5,rm=0,hollow,incolor=Goldenrod](0,4,0)
\psSolid[object=plan,action=draw,definition=equation,
  args={[0.707 0 0.707 0] 90},base=-2 2 -3 3,planmarks]
\psSolid[object=line,args=0 0 0 5.5 0,
  linecolor=blue]% first half of y axis
\color{white}\axesIIIID[showOrigin=false,
  linecolor=blue](0,6.8,0)(3.5,8,3.5)
\end{pspicture}
```

1.1 The PSTricks-Project

The first version of the main package `pstricks` from *Timothy Van Zandt* was published nearly 20 years ago and is still the base package for the so-called PSTricks project. A list of all additional packages which are published since 1991 are listed at the PSTricks web page <http://PSTricks.tug.org>. In [1] and especially [2] the packages are described and shown with a lot of examples. Here, we will list the packages with only one significant example, which may show what the package provides. More examples or some more information can be found on the PSTricks web page (<http://PSTricks.tug.org>), CTAN (<http://mirrors.ctan.org>) or at the local \TeX distribution by calling the documentation with the `texdoc` program.

All PSTricks packages load by default the main package `pstricks`, which itself loads the package `xcolor`, which has a better support for colors than the package `color`.

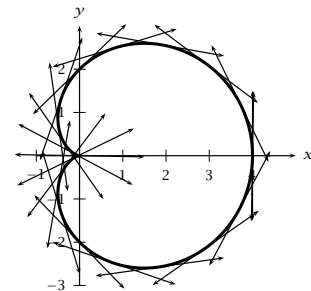
pstricks: Main package with the base macros for lines, curves, areas, etc.



```
\usepackage{pstricks}
```

```
\psset{unit=0.05mm}% 1:20, or=0.1mm for 1:10
\begin{pspicture}(-100,-100)(1000,650)
\pspolygon[linewidth=2pt,fillcolor=gray!10,
  fillstyle=solid](0,0)(0,470)(860,625)(860,0)
\psset{linewidth=0.2pt,arrowscale=2,tbar=10pt}
\psline{|<->|}(0,-100)(860,-100)
\rput*(430,-100){430mm}
\psline{|<->|}(960,0)(960,625)
\rput*(90){960,312.5}{625mm}
\psline{|<->|}(-100,0)(-100,470)
\rput*(90){(-100,235){470mm}}
\end{pspicture}
```

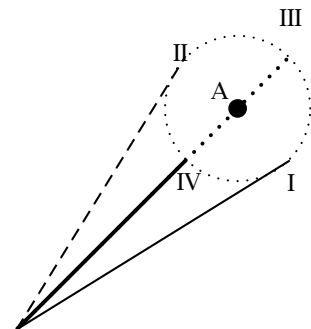
pstricks-add: Extended base macros for the packages `pstricks`, `pst-node`, and `pst-plot`



```
\usepackage{pstricks-add}
\usepackage{multido}

\psscalebox{0.75}{%
\begin{pspicture}(-1,-3)(4.75,3)
\psaxes{>->}(0,0)(-1,-3)(5,3)
\psplot[polarplot,linewidth=2pt,algebraic,
  plotpoints=500]{0}{6.289}{2*(1+cos(x))}
\multido{\r=0.000+0.314}{21}{%
\psplotTangent[polarplot,Derive=-2*sin(x),
  algebraic,arrows=<->]{\r}{1.5}{2*(1+cos(x))}}
\end{pspicture}}
```

pst-node: Nodes and node connections in text and a matrix

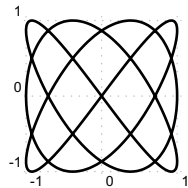


```
\usepackage{pst-node}
\SpecialCoor

\begin{Example}[ltxps]{\xLcs{pnode}\xLcs{uput}}
\begin{pspicture}(4,4)
\pnode(3,3){A}\psdot[dotstyle=2](A)\uput[135](A){A}
\pscicle[linestyle=dotted](A){1}
\end{pspicture}
\end{Example}
```

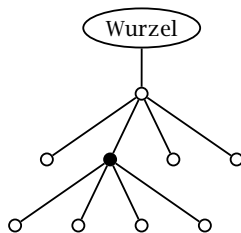
```
\psline([nodesep=1,angle=-45]A)\uput[0](3.5,2){I}
\psline[linestyle=dashed]([nodesep=-1,angle=-45]A)
\uput[-45](2,4){II}
\psline[linestyle=dotted,linewidth=1.5pt]%
([offset=1,angle=-45]A)\uput[-225](4,4){III}
\psline[linewidth=1.5pt]([offset=1,angle=135]A)
\uput[0](2,2){IV} \ncurve{>}{A}{0,0}
\end{pspicture}
```

pst-plot: Plotting of mathematical functions or external data sets



```
\usepackage{pst-plot}
\psset{xunit=1.5cm,yunit=1.5cm}
\begin{pspicture}[showgrid=true](-1.1,-1.1)(1.1,1.1)
\psparametricplot[plotstyle=curve,linewidth=1.5pt,
plotpoints=200]{-360}{360}%
{t 1.5 mul sin t 2 mul 60 add sin}
\end{pspicture}
```

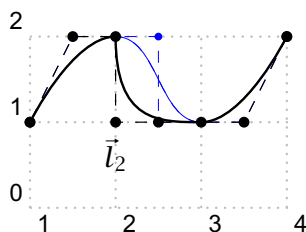
pst-tree: Trees



```
\usepackage{pst-tree}
\pstree[levelsep=1cm,radius=3pt]{\Toval{Wurzel}}{%
\TC
\pstree{\TC*}{\TC\TC\TC\TC}%
\TC\TC}}

```

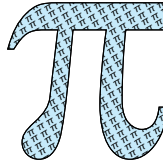
pst-bezier: Bézier curves



```
\usepackage{pst-bezier}
\pspicture[showgrid=true](5,3)\psset{showpoints=true}
\psbcurve[linestyle=blue,linewidth=0.01](1,1)%
(2,2)(3,1)(4,2)
```

```
\psbcurve(1,1)(2,2)\l(2,1)(3,1)(4,2)
\uput[-90](2,1){$\vec{l}_2$}
\end{pspicture}
```

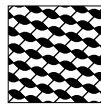
pst-text: Character and text manipulation



```
\usepackage{pst-text}
\usepackage[tiling]{pst-fill}
\DeclareFixedFont{\ps}{U}{psy}{m}{n}{8cm}

\psboxfill{\footnotesize$\pi$}
\begin{pspicture}(0,0)(.25\linewidth,8)
\pscharpath[fillstyle=solid,fillcolor=cyan!20,
addfillstyle=boxfill,fillangle=30,
fillsep=0.6mm]{%
\rput[b](-0.5,0){\ps\char112}}
\end{pspicture}
```

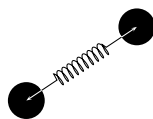
pst-fill: Filling and tiling



```
\usepackage[tiling]{pst-fill}
\newcommand\FArcW{%
\begin{pspicture}(-0.25,-0.25)(0.25,0.25)%
\pswedge*(-0.25,-0.25){0.25}{0}{90}
\pswedge*(0.25,0.25){0.25}{180}{270}
\psframe[linewidth=0.1pt]%
(-0.25,-0.25)(0.25,0.25)
\end{pspicture}}

\begin{pspicture}(3.1,3.1) \psboxfill{\FArcLW}
\psframe[fillstyle=boxfill,fillcyclex=2,
fillangle=45](3,3)
\end{pspicture}
```

pst-coil: Coils and zigzag lines



```
\usepackage{pst-node,pst-coil} \SpecialCoor
\begin{pspicture}(4,3)
\cnode*(0.5,0.5){0.5}{A}
\cnode*(3.5,2.5){0.5}{B}
\pccoil[coilwidth=0.4,
coilaspect=35,coilheight=0.5,
linecolor=white]{<->}(A)(B)
\nccoil[coilwidth=0.4,coilaspect=35,
coilheight=0.5]{A}{B}
\end{pspicture}
```

pst-grad: Color gradients



```
\usepackage{pst-grad}
\begin{pspicture}(3,2.25)
\psframe[fillstyle=gradient](3,2)
\end{pspicture}
```


pst-slope: Extended color gradients



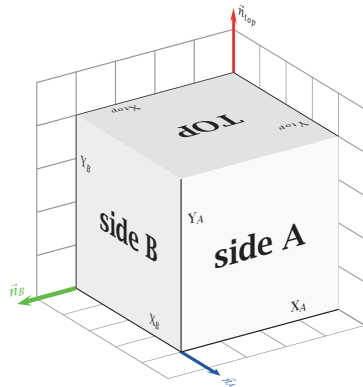
```
\usepackage{pst-slope}
\def\st#1{\makebox[2.75cm]{$\vcenter to55pt{\#1$}}
\begin{pspicture}(2.5,2.5)
\psset{fading,endfading=0.75,linewidth=black!40}
\psframe*(-0.3,-0.25)(3.5,20pt)
\psframebox[fillstyle=slope]{\LARGE\st{slope}}
\end{pspicture}
```

pst-blur: Shadows



```
\usepackage{pst-blur,pst-text}
\DeclareFixedFont{\RM}{T1}{ptm}{b}{n}{1.75cm}
\psset{shadow=true,blur=true,shadowsize=10pt,
blurradius=5pt}
\pscharpath{\RM PSTricks}
```

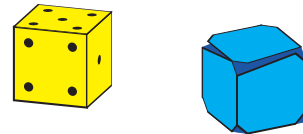
pst-3d: Basic macros for tilting and three dimensional objects



```
\usepackage{pst-3d}
\begin{pspicture}(-4.5,-3)(3,4.75)
\psset{viewpoint=1 1.5 0.8}
{\psset{gridlabels=0pt,subgriddiv=0,gridcolor=black!50}
\ThreeDput[normal=0 0 1]{\psgrid(5,5)}
\ThreeDput[normal=0 -1 0]{\psgrid(5,5)}
\ThreeDput[normal=1 0 0]{\psgrid(5,5)}
\ThreeDput[normal=0 0 1]{%
\psline[linewidth=3pt,linewidth=blue]{->}(4,4)(4,5)
\uput[90](4,5){%
\psrotateleft{\textcolor{blue}{$\vec{n}_A$}}}{%$xy
\ThreeDput[normal=0 -1 0]{%
\psline[linewidth=3pt,linewidth=green]{->}(4,0)(5,5,0)
```

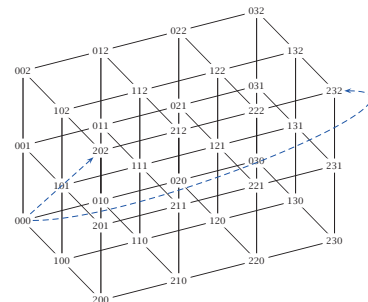
```
\uput[90](5.5,0){\psscalebox{-1 1}{%
\textcolor{green}{$\vec{n}_B$}}}{%$xz
\ThreeDput[normal=1 0 0]{%
\psline[linewidth=3pt,linewidth=red]{->}(0,4)(0,5)
\uput[0](0,5){$\vec{n}_-{\text{top}}$}} %yz
\ThreeDput[normal=0 0 1](0,0,4){%
\psframe*[linewidth=gray!25](4,4)
\rput(2,2){\Huge\textbf{TOP}}}}
\ThreeDput[normal=0 1 0](4,4,0){%
\psframe*[linewidth=gray!5](4,4)
\rput(2,2){\Huge\textbf{side A}}}}
\ThreeDput[normal=1 0 0](4,0,0){%
\psframe*[linewidth=gray!15](4,4)
\rput(2,2){\Huge\textbf{side B}}}}
% Die kleinen Achsen
\ThreeDput[normal=0 0 1](0,0,4){%
\psline(4,0)\uput[90](3,0){X$_-{\text{top}}$}
\psline(0,4)\uput[0](0,3){Y$_-{\text{top}}$}}
\ThreeDput[normal=0 1 0](4,4,0){%
\psline(4,0)\uput[90](3,0){X$_-{\text{A}}$}
\psline(0,4)\uput[0](0,3){Y$_-{\text{A}}$}}
\ThreeDput[normal=1 0 0](4,0,0){%
\psline(4,0)\uput[90](3,0){X$_-{\text{B}}$}
\psline(0,4)\uput[0](0,3){Y$_-{\text{B}}$}}
\end{pspicture}
```

pst-ob3d: Simple three dimensional objects



```
\usepackage{pst-ob3d}\SpecialCoor
\begin{pspicture}(-1,-0.5)(3,1.7)
\psset{fillstyle=solid,fillcolor=yellow,RandomFaces=true}
\PstDie[viewpoint=1 -3 1]
\rput(1.5,0){\PstCube[Corners=true,CornersColor=blue,
fillstyle=solid,fillcolor=cyan,viewpoint=1 2 1]{1}{1}{1}}
\end{pspicture}
```

pst-gr3d: Simple three dimensional grids



```
\usepackage{pst-gr3d,pst-node,multido}\SpecialCoor
\psscalebox{0.5}{\footnotesize%
\PstGridThreeD[GridThreeDNodes,unit=2.5](2,3,2)
\multido{\ix=0+1}{3}{%
\multido{\iy=0+1}{4}{%
```

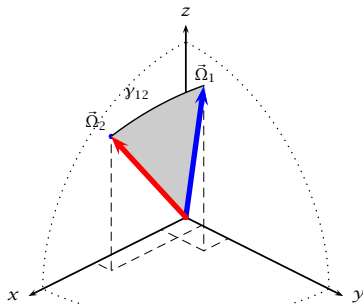
```
\multido{\iz=0+1}{3}{%
  \rput*(Gr3dNode\ix\iy\iz){$\ix\iy\iz$}}
\psset{linecolor=blue,linestyle=dashed,
  linewidth=0.3pt,arrowscale=2,nodesep=8pt}
\pcline{>}(Gr3dNode000)(Gr3dNode202)
\pccurve{>}(Gr3dNode000)(Gr3dNode232)}
```

pst-fr3d: Three dimensional buttons

Off **On**

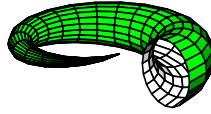
```
\usepackage{pst-fr3d}
\PstFrameBoxThreeD[FrameBoxThreeDOn=false]%
{\Large Off}
\quad\PstFrameBoxThreeD{\Large On}% the default
```

pst-3dplot: Three dimensional graphic objects in parallel projection



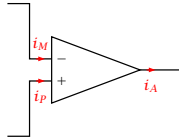
```
\usepackage{pst-3dplot}
\def\oA{\pstThreeDLine[linecolor=blue,linewidth=3pt,
  arrows=c->](0,0,0)(1,60,70)}
\def\oB{\pstThreeDLine[linecolor=red,linewidth=3pt,
  arrows=c->](0,0,0)(1,10,50)}
\def\oAB{\pstThreeDEllipse[beginAngle=58,
  endAngle=90](0,0,0)(1,140,40)(1,10,50)}
\begin{pspicture}(-4.8,-1.75)(4.8,3.75)
\psset{unit=4cm,drawCoor,beginAngle=90,endAngle=180,
  linestyle=dotted}
\pstThreeDCoor[drawing,linewidth=1pt,linecolor=black,%
  linestyle=solid,xMin=0,xMax=1.1,yMin=0,yMax=1.1,
  zMin=0,zMax=1.1]
\pstThreeDEllipse(0,0,0)(-1,0,0)(0,1,0)
\pstThreeDEllipse(0,0,0)(-1,0,0)(0,0,1)
\pstThreeDEllipse[beginAngle=0,
  endAngle=90](0,0,0)(0,0,1)(0,1,0)
\psset{SphericalCoor,linestyle=solid}
\pstThreeDDot[dotstyle=none](1,10,50)
\pstThreeDDot[dotstyle=none](1,60,70)
\pscustom[fillstyle=solid,fillcolor=black!20,
  linestyle=none]{\oB\oAB\oA} \oA\oB\oAB
\pstThreeDPut[origin=lb](1.1,60,70){$\vec{\Omega}_1$}
\pstThreeDPut[origin=rb](1.2,10,50){$\vec{\Omega}_2$}
\pstThreeDPut[origin=lb](1,10,65){$\gamma_{12}$}
\end{pspicture}
```

pst-solides3d: Three dimensional graphic objects in central projection



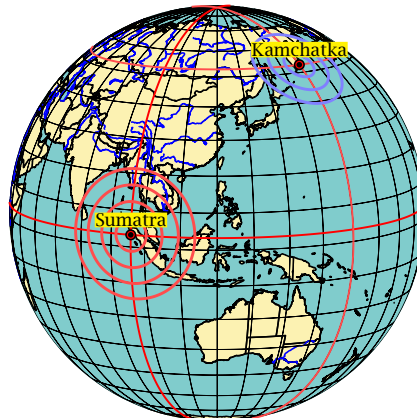
```
\usepackage{pst-solides3d}
\psset{viewpoint=100 50 20 rtp2xyz,
  Decran=60,lightsrc=10 15 10}
\defFunction[algebraic]{corne}(u,v)%
{(2 + u*cos(v))*sin(2*pi*u)}%
{(2 + u*cos(v))*cos(2*pi*u)+2*u}%
{u*sin(v)}
\begin{pspicture}(-2,-2)(2,2)
\psSolid[object=surfaceparametree,
  base=0 1 0 2 pi mul,
  function=corne,ngrid=20]
\end{pspicture}
```

pst-circ: Electrical circuits for electronic and microelectronic



```
\usepackage{pst-circ}
\begin{pspicture}(4,3.5)
\node(0,3){A}\node(0,0){B}
\node(4,1.5){C}
\OA[OAperfect=false,OAiplus,
  OAipluslabel=$i_P$,
  OAipluslabel=$i_M$,
  OAioutlabel=$i_A$,
  intensitycolor=red,
  intensitylabelcolor=red](A)(B)(C)
\end{pspicture}
```

pst-geo: Two and three dimensional geographical objects



```
\usepackage{pst-map3d}
\psset{unit=0.75,Radius=5,Dobs=200,Decran=200,
  path=data/pstricks,PHI=10,THETA=120,circlewidth=1.5pt}
\begin{pspicture}(-5,-5)(5,5)
\WorldMapThreeD[circles=false,australia=true]
\psmeridien{95.98} \psparallel{3.30}
\psepicer[circlecolor=red!70,waves=4,
```

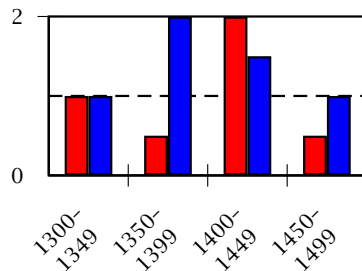
```
Rmax=2000](95.98,3.30){Sumatra}
\psmeridien[meridiencolor=red!70]{160}
\psparallel[parallelcolor=red!70]{52.76}
\psepicercenter[circlearcolor=blue!50](160,52.76){Kamchatka}
\end{pspicture}
```

pst-barcode: Barcodes



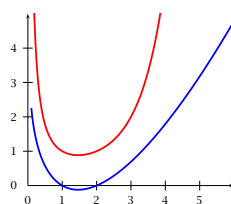
```
\usepackage{pst-barcode}
\begin{pspicture}(lin,lin)
\psbarcode{%
Herbert Voss Wasgenstraße 21 14129 Berlin
http://www.dante.de/}%
{rows=52 columns=52}{datamatrix}
\end{pspicture}
```

pst-bar: Bar diagrams



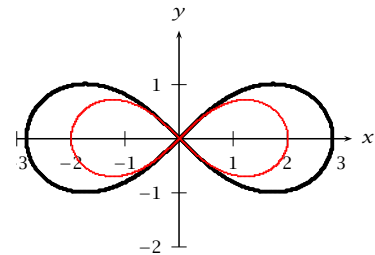
```
\usepackage{pst-plot,pst-bar}
\begin{filecontents*}{data1T.csv}
1300--1349, 1350--1399, 1400--1449, 1450--1499
1, 0.5, 2, 0.5
1, 2, 1.5, 1
\end{filecontents*}
\readpsbardata{\data}{data1T.csv}
\begin{pspicture}(-0.5,-2)(4,2)\footnotesize
\psline[linestyle=dashed](0,1)(4,1)
\psaxes[axesstyle=frame,Dy=2,labels=y](0,0)(4,2)
\psbarchart[barstyle={red,blue},barlabelrot=45,
chartstyle=cluster]{\data}
\end{pspicture}
```

pst-math: Extended PostScript functions



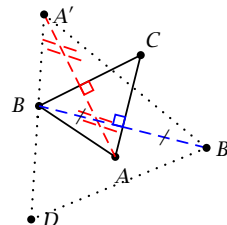
```
\usepackage{pst-plot,pst-math}
\begin{pspicture*}(-0.75,-.75)(6,5)
\psaxes{->}(6,5)
\psset{linewidth=1.5pt,
plotpoints=200}
\psplot{.1}{6}{x GAMMA}
\psplot{.1}{6}{x GAMMALN}
\end{pspicture*}
```

pst-func: Special mathematical functions like polynomials, distributions, implicit defined functions, etc.



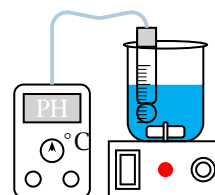
```
\usepackage{pst-func,pstricks-add}
\begin{pspicture*}(-3,-2.2)(3.6,2.5)
\psaxes{->}(0,0)(-3,-2)(3.2,2)[x$,0][$y$,90]
\psplotImp[linewidth=2pt,algebraic](-5,-2.2)(5,2.4){
(x^2+y^2)^2-8*(x^2-y^2)}
\rput*(2,1.5){$\left(x^2+y^2\right)^2-8(x^2-y^2)=0$}
\psplotImp[linewidth=1pt,linestyle=red,
algebraic](-5,-2.2)(5,2.4){(x^2+y^2)^2-4*(x^2-y^2)}
\end{pspicture*}
```

pst-eucl: Euklidian geometry



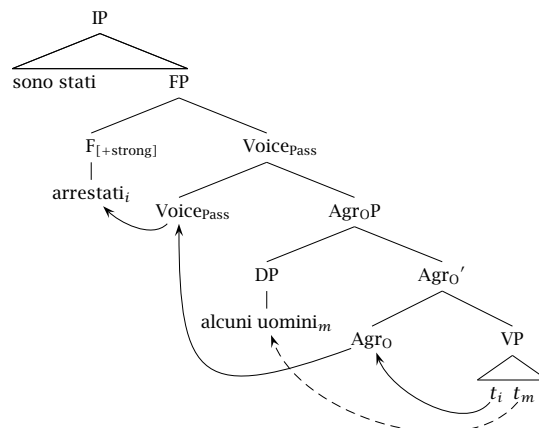
```
\usepackage{pst-eucl}
\psset{unit=0.4}%
\begin{pspicture}(0,-2)(8,7)
\pstTriangle(1,3){B}(5,5){C}(4,1){A}
\pstOrtSym{A}{B}{C}{D}
\psset{CodeFig=true}
\pstOrtSym[CodeFigColor=red]{C}{B}{A}
\pstOrtSym[SegmentSymbol=pstslash,
dotsep=3mm,linestyle=dotted,
CodeFigColor=blue]{C}{A}{B}
\pspolygon[linestyle=dotted,
linewidth=1pt](A')(B')(D)
\end{pspicture}
```

pst-labo: Chemical objects



```
\usepackage{pst-labo}
\psset{unit=0.5cm,
glassType=becher,
burette=false}
\pstDosage[pHmetre]
```

pst-jtree: Linguistic trees



```

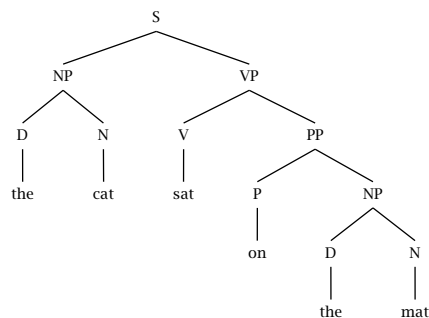
\usepackage{pst-jtree}

\jtree[xunit=5em,yunit=2em]

\! = {IP}

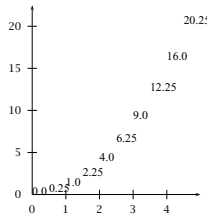
<tri>{\triline{sono stati\hfil}}
^<tri>[triratio=.95]{FP}
:{F$_{\rlap{$\scriptstyle\rm
[+strong]$}}$}!a {Voice$_{\%
\rlap{$\scriptstyle\rm Pass$}}$}
:{Voice\rlap{$_{\rm Pass}$}}@A2 {$\rm Agr\_OP$}
:{DP}!b {$_{\rm Agr\_0}$}'$}
:[scaleby=.8 1]{$\rm Agr\_0$}@A3 [scaleby=.8 1]{VP}
<tri>[scaleby=.4 .7]{\rnode{A5}{\t_i$}
\hskiplex \rnode{A6}{\t_m$}}.
\!a = <shortvert>{\arrestati_i_$}@A1 .
\!b = <shortvert>{\alculi uomini_m_$}@A4 .
\psset{arrows=->}
\nccurve[angleA=225,angleB=-45]{A2}{A1}
\nccurve[angleA=200,angleB=-90,ncurv=1.5]{A3}{A2}
\nccurve[angleA=-130,angleB=-70]{A5}{A3}
\nccurve[angleA=-130,angleB=-70,
linestyle=dashed]{A6}{A4}
\enditree

```



```
\usepackage{pst-qtree}
\begin{Example}[ltxps]{\xLcs{pst-qtree}{Tree}}
\Tree
[.S [.NP [.D the ] [.N cat ]
] [.VP [.V sat ] [.PP [.P on ]
[.NP [.D the ] [.N mat ] ] ] ] ] ]
```

infix-RPN: Converting an algebraic expression (infix) in a PostScript expression (postfix)



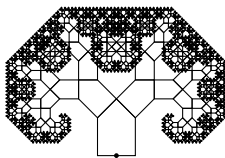
```
\usepackage{infix-RPN,pst-func,
multido}
\SpecialCoor
\psset{yunit=0.25}
\begin{pspicture}(-0.25,-2)(5,22.5)
\infixtoRPN{x*x}
\multido{\rx=0.0+0.5}{10}{%
\rput{!/x \rx\space def
\RPN\space x exch }{%
\psPrintValue{\RPN}}}
\psaxes[dy=5,Dy=5]{->}(5,22.5)
\end{pspicture}
```

pst-lens: Lens magnification

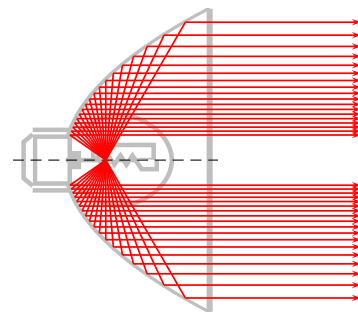


```
\usepackage{pst-lens}
\def\Wishes{%
\rput[lb](0,0){%
\Large\begin{minipage}{3cm}
\centering
\textbf{Alles Gute}\!
\textbf{Jana},\!
all the best for
the year\!Huge 2010!
\end{minipage}}}%
\begin{pspicture}(0,-1.5)(3,4)
\Wishes\PstLens[LensMagnification=2]%
(1.5,2.5){\Wishes}
\end{pspicture}
```

pst-fractal: Fractals

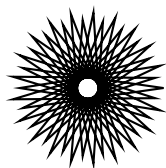


```
\usepackage{pst-fractal}
\begin{pspicture}(-3,0)(3,4)
\psPTree\psdot*(0,0)
\end{pspicture}
```



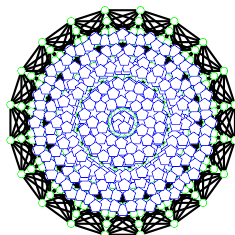
pst-optic: Two dimensional optical arrangement

pst-poly: Polygons



```
\usepackage{pst-poly}
\PstPolygon[PolyNbSides=21,
PolyOffset=2,
PolyIntermediatePoint=-0.9]
```

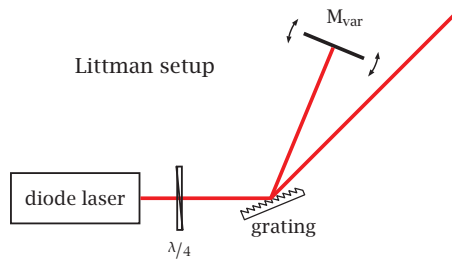
pst-coxeterp: Regular polytopes



```
\usepackage{pst-coxeterp}
\begin{pspicture}(-2,-2)(2,2)
\psset{unit=0.4cm,
colorCenters=blue,
styleCenters=pentagon,
sizeCenters=0.2}
\gammapn[P=5,dimension=4]
\end{pspicture}
```

```
\usepackage{pst-optic}
\psset{unit=0.5}
\begin{pspicture}(-1.5,-5.5)(10,5.5)
\rput(0,0){\beamLight[drawing=false,mirrorDepth=4.75,
mirrorWidth=0.1,mirrorHeight=10,linecolor=lightgray]}
\makeatletter \pst@getcoor{Focus}\pst@tempf
\psset{linecolor=red}
\multido{\n=60+5}{18}{\mirrorCVGRay[linecolor=red,
mirrorDepth=4.75,mirrorHeight=10](Focus)(!
/XF \pst@tempf pop \pst@number\psxunit div def
\n\space cos XF add \n\space sin neg){Endd1}
\psOutLine[arrows=->,length=.25](Endd1)(Endd1')}{Endd2}
\mirrorCVGRay[linecolor=red,mirrorDepth=4.75,
mirrorHeight=10](Focus)(!
/XF \pst@tempf pop \pst@number\psxunit div def
\n\space cos XF add \n\space sin )}{End1}
\psOutLine[arrows=->,length=.25](End1)(End1')}{End2}
\makeatother
\end{pspicture}
```

pst-optexp: Experimental optic



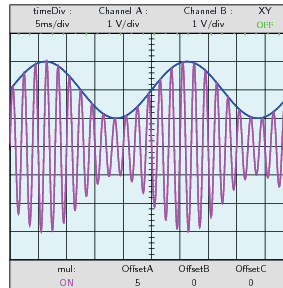
```
\usepackage{pst-optexp, nicefrac}
\begin{pspicture}(-4,-1)(3,3)
\addtopsstyle{Beam}{linewidth=2\pslinewidth,
  linecolor=red!90!black}
\psset{labeloffset=0.5}
\node(-2,0){LaserOut}\node(0,0){Grat}
\node(4;45){Out}\node(2.5;67.5){Mvar}
\optbox[optboxwidth=2,labeloffset=0,
  endbox](Grat)(LaserOut){diode laser}
\mirror[variable,conn=o-] %
  (Grid)(Mvar)(Grid){M$_\mathrm{var}$}
\optgrid[beam](LaserOut)(Grat)(Out){grating}
\optretplate[position=0.3,labeloffset=0.8] %
  (LaserOut)(Grat){$\nicefrac{\lambda}{4}$}
\rput[l](-3,2){Littman setup}
\end{pspicture}
```

pst-diffraction: Generating of a diffraction



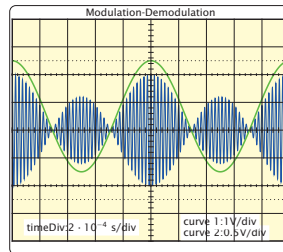
```
\usepackage{pst-diffraction}
\begin{pspicture}(-3.5,-1.5)(3.5,3.5)
\psdiffractionCircular[IIID,r=0.5e-3,f=10,
  pixel=0.5,lambda=520,colorMode=0]
\end{pspicture}
```

pst-osci: Simulating the output of an oscilloscope



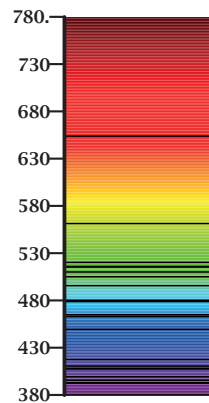
```
\usepackage{pst-osci}
\Oscillo[amplitude1=1,
  amplitude2=1,CC2=2,
  period2=25,period1=2,
  combine=true,
  operation=mul,
  offset1=5]
```

pst-am: Modulation and demodulation



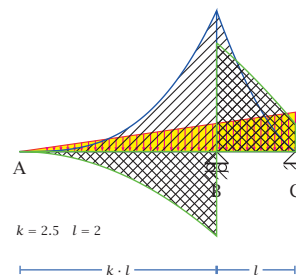
```
\usepackage{pst-am}
\psAM[SignalModule,
  SignalModule,
  timeDiv=2e-4,
  U0=0.5,
  frequencePorteuse=4e4,
  Up=4,Um=2,
  voltDivY2=0.5,values]
```

pst-spectra: Spectral lines



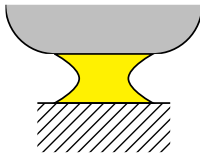
```
\usepackage{pst-spectra,
  psticks-add}
\begin{pspicture}(0,-0.75)(2,4.75)
\rput{90}(1,0){%
  \psspectrum[axe,Dl=50,wlangle=-90,
    wlcmd={\scriptsize\bfseries},
    begin=780,end=380,
    element=Es,
    absorption](5,2)(0,0)}
\end{pspicture}
```

pst-stru: Structural schemes in civil engineering



```
\usepackage{pst-stru}
\begin{pspicture}(-1,-3)(12,4)
\psset{arrowsize=0.8mm,
  arrowinset=0}
\triload[K=2.5,P=8,L=2]
\end{pspicture}
```

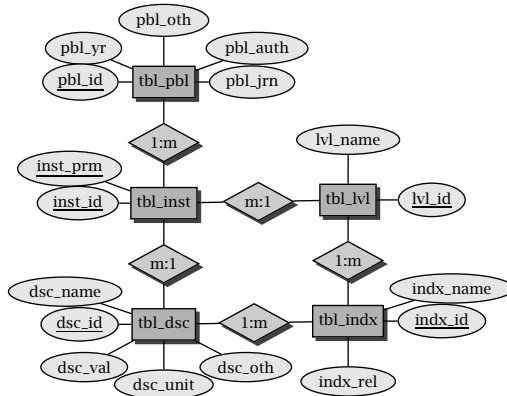

pst-pad: Attachment models



```
\usepackage{pst-pad}

\begin{pspicture}(4,4)
\PsPad(2,2)
\end{pspicture}
```

pst-dbicons: Entity-Relationship diagrams



```

\usepackage{pst-dbicons}

\seticonparams{entity}{shadow=true,fillcolor=black!30,
  fillstyle=solid}

\seticonparams{attribute}{fillcolor=black!10,
  fillstyle=solid}

\seticonparams{relationship}{shadow=true,
  fillcolor=black!20,fillstyle=solid}

\begin{tabular}{cc}
  \begin{tabular}{c}
    \entity{tblpbl}[tbl\_pbl]\\[2cm]
    \entity{tblinst}[tbl\_inst]\\[2cm]
    \entity{tbldsc}[tbl\_dsc] \\[2cm]
  \end{tabular}\hspace{6em}
  \begin{tabular}{c}-\\[2cm]
    \entity{tbllvl}[tbl\_lvl]\\[2cm]
    \entity{tblindx}[tbl\_indx]\\[2cm]
  \end{tabular}
\end{tabular}

\attributeof{tblpbl}[3em]{0}{pbljrn}[pbl\_jrn]
\attributeof{tblpbl}[3em]{90}{pbloth}[pbl\_oth]
\attributeof{tblpbl}[3em]{30}{pblauth}[pbl\_auth]
\attributeof{tblpbl}[3em]{150}{pblyr}[pbl\_yr]
\attributeof{tblpbl}[3em]{180}[key]{pblid}[pbl\_id]
\attributeof{tblinst}[3em]{150}[key]{instprm}[inst\_prm]
\attributeof{tblinst}[3em]{180}[key]{instid}[inst\_id]
\attributeof{tbldsc}[3em]{180}[key]{dscid}[dsc\_id]
\attributeof{tbldsc}[3em]{150}{dscname}[dsc\_name]
\attributeof{tbldsc}[3em]{220}{dscval}[dsc\_val]
\attributeof{tbldsc}[3em]{270}{dscunit}[dsc\_unit]
\attributeof{tbldsc}[3em]{320}{dscoth}[dsc\_oth]
\attributeof{tbllvl}[3em]{0}[key]{lvlid}[lvl\_id]
\attributeof{tbllvl}[3em]{90}{lvlname}[lvl\_name]
\attributeof{tblindx}[3em]{0}[key]{indxid}[indx\_id]
\attributeof{tblindx}[3em]{30}{indxname}[indx\_name]
\attributeof{tblindx}[3em]{270}{indxrel}[indx\_rel]
\relationshipbetween{tblpbl}{tblinst}{1:m}%relationship

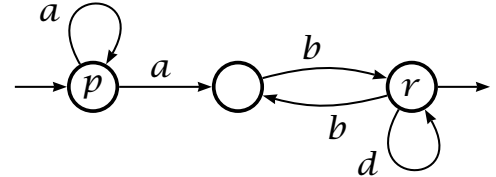
```

```

\relationshipbetween{tblinst}{tblnsc}{m:1}
\relationshipbetween{tblinst}{tblnvl}{m:1}
\relationshipbetween{tblnsc}{tblindx}{1:m}
\relationshipbetween{tblnvl}{tblindx}{1:m}
\end{tabular}

```

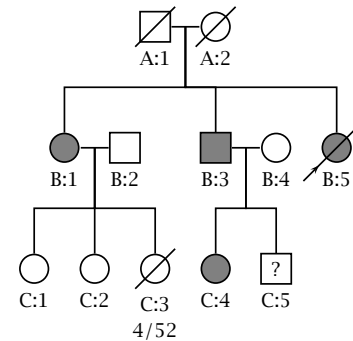
pst-vaucanson-g: Drawing automata and graphs



```
\usepackage{vcaucanson-g}

\begin{VCPicture}{(0,-2)(5.5,2)}
\State[p]{(0,0)}{A} \State{(2.5,0)}{B}
\State[r]{(5.5,0)}{C} \Initial{A} \Final{C}
\EdgeL{A}{B}{a} \ArcL{B}{C}{b}
\ArcL{C}{B}{b} \LoopN{A}{a} \LoopS{C}{d}
\end{VCPicture}
```

pst-pdgr: "Pedegrees", medical structures



```

\usepackage{pst-pdgr}

\begin{pspicture}(6,6) \psset{belowtextrp=t,armB=1}
\rput(2.5,5.5){\pstPerson[male,deceased,
belowtext=A:1]{A:1}}
\rput(3.5,5.5){\pstPerson[female,deceased,
belowtext=A:2]{A:2}}
\pstRelationship[descentnode=A:1.2]{A:1}{A:2}
\rput(1,3.5){\pstPerson[female,affected,
belowtext=B:1]{B:1}}
\pstDescent{A:1.2}{B:1}
\rput(2,3.5){\pstPerson[male,belowtext=B:2]{B:2}}
\pstRelationship[descentnode=B:1.2]{B:1}{B:2}
\rput(3.5,3.5){\pstPerson[male,affected,
belowtext=B:3]{B:3}}
\pstDescent{A:1.2}{B:3}
\rput(4.5,3.5){\pstPerson[female,belowtext=B:4]{B:4}}
\pstRelationship[descentnode=B:3.4]{B:3}{B:4}
\rput(5.5,3.5){\pstPerson[female,affected,deceased,
proband,belowtext=B:5]{B:5}}
\pstDescent{A:1.2}{B:5}
\rput(0.5,1.5){\pstPerson[female,belowtext=C:1]{C:1}}

```

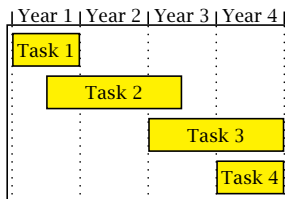
```
\pstDescent{B:1.2}{C:1}
\rput(1.5,1.5){\pstPerson[female,belowtext=C:2]{C:2}}
\pstDescent{B:1.2}{C:2}
\rput(2.5,1.5){\pstPerson[female,deceased,
  belowtext={\tabular{c}{C:3\4/52\endtabular}}{C:3}}
\pstDescent{B:1.2}{C:3}
\rput(3.5,1.5){\pstPerson[female,affected,
  belowtext=C:4]{C:4}}
\pstDescent{B:3.4}{C:4}
\rput(4.5,1.5){\pstPerson[male,insidetext=?,
  belowtext=C:5]{C:5}}
\pstDescent{B:3.4}{C:5}
\end{pspicture}
```

pst-light3d: Three dimensional light effects



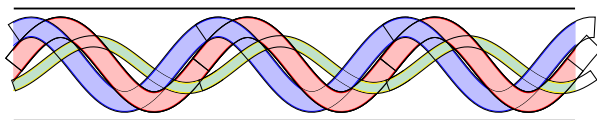
```
\usepackage{pst-light3d}
\DeclareFixedFont{\RM}{T1}{ptm}{m}{n}{2cm}
\psset{linestyle=none,fillstyle=solid,
  fillcolor={\rgb{1,0.84,0}}}
\PstLightThreeDText[LightThreeDLength=0.5]{\RM\TeX}
```

pst-gantt: Gantt charts



```
\usepackage{pst-gantt}
\begin{PstGanttChart}[yunit=1.5,
  ChartUnitIntervalName=Year,
  ChartUnitBasicIntervalName=Month,
  TaskUnitIntervalValue=12,TaskUnitType=Year,
  ChartShowIntervals]{4}{4}
\PstGanttTask[TaskInsideLabel={Task 1}]{0}{1}
\PstGanttTask[TaskInsideLabel={Task 2},
  TaskUnitType=Month]{6}{24}% 24 mon start at 6
\PstGanttTask[TaskInsideLabel={Task 3}]{2}{2}
\PstGanttTask[TaskInsideLabel={Task 4}]{3}{1}
\end{PstGanttChart}
```

pst-thick: Very thick lines and curves



```
\usepackage{pst-thick}
\newpsstyle{thicklinejaune}{fillstyle=solid,
  fillcolor=yellow!50!cyan!50,linewidth=yellow,
  plotpoints=360}
\newpsstyle{thicklinevert}{fillstyle=solid,
  fillcolor=green!50,linewidth=green,plotpoints=360}
\newpsstyle{onlycurvejaune}{linewidth=yellow,plotpoints=360}
```

```
\def\SinusPhase#1#2#3{%
  /P #1 def /A #2 def /F #3 DegtoRad def
  /O 360 P div def /x0 t def
  /y0 t F add 0 mul sin A mul def % A*sin(O*t)
  /dx dt def /dy t F add dt add 0 mul sin
  t F add 0 mul sin sub A mul def }

\psset{unit=0.5}
\begin{pspicture}(0,-4)(30,4)
\def\motif{\psclip{\psframe[linestyle=none,
  dimen=inner](0,-3)(10,3)}
\pstthick[stylethick=thicklineblue]{-1}{11}%
  {\SinusPhase{10}{2}{90}}
\pstthick[stylethick=thicklinejaune,E=0.5]{-1}{11}%
  {\SinusPhase{10}{1.25}{-100}}
\pstthick{-1}{11}{\SinusPhase{10}{2}{0}}
\psclip{\pstthick[stylethick=vide,E=1.1]%
  {-1}{11}{\SinusPhase{10}{2}{0}}}
\pstthick[stylethick=thicklineblue]{0}{3}%
  {\SinusPhase{10}{2}{90}}
\endpsclip%
\psclip{\pstthick[stylethick=vide,E=1.1]{0}{11}%
  {\SinusPhase{10}{2}{90}}}
\pstthick{5}{9}{\SinusPhase{10}{2}{0}}
\endpsclip%
\psclip{\pstthick[stylethick=vide,E=0.6]{0}{11}%
  {\SinusPhase{10}{1.25}{-100}}}
\pstthick[stylethick=thicklineblue]{7}{9}%
  {\SinusPhase{10}{2}{90}}
\endpsclip%
\psclip{\pstthick[stylethick=vide,E=1.1]{0}{10}%
  {\SinusPhase{10}{2}{0}}}
\pstthick[stylethick=thicklinejaune,E=0.5]{7}{11}%
  {\SinusPhase{10}{1.25}{-100}}
\endpsclip%
\psclip{\pstthick[stylethick=vide,E=1.1]{0}{11}%
  {\SinusPhase{10}{2}{0}}}
\pstthick[stylethick=thicklinejaune,E=0.5]{-0.5}{1}%
  {\SinusPhase{10}{1.25}{-100}}
\endpsclip%
\endpsclip%
\motif\rput(10,0){\motif}\rput(20,0){\motif}
\psline[linewidth=0.1](0,3)(30,3)
\psline[linewidth=0.1](0,-3)(30,-3)
\end{pspicture}
```

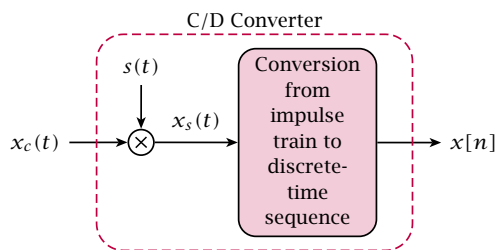
pst-mirror: Projections on a spherical mirror



```
\usepackage{pst-mirror,pst-grad}
\psscalebox{0.7}{\begin{pspicture}(-7,-7)(7,7)
\newpsstyle{GradWhiteYellow}{fillstyle=gradient,
  gradbegin=yellow,gradend=yellow!20,linewidth=yellow!50,
```

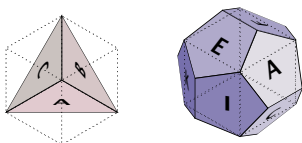
```
GradientCircle=true,gradmidpoint=0,GradientPos={{(1,1)}}%
\pscircle[style=GradWhiteYellow]{7.07}
\pstSphereGrid[linecolor=red,grille=10,Ymin=-50,Ymax=50,
  Xmax=80,Xmin=-80,normale=0 0](20,0,0)
\pstSphereGrid[linecolor=blue,grille=10,Ymin=-40,Ymax=-20,
  Xmax=80,Xmin=-80,normale=0 90](40,0,-10)%
\pstTextSphere[fillstyle=solid,fillcolor=red,normale=0 0,
  fontsize=40,PSfont=Time-Roman,y0=0](20,0,10){pst-mirror}
\pstTextSphere[fillstyle=solid,fillcolor=black,normale=0 0,
  fontsize=20,PSfont=Helvetica,y0=0](20,0,35){PSTricks}
\pstTextSphere[fillstyle=solid,fillcolor=blue,normale=0 90,
  fontsize=10,PSfont=Helvetica,
  y0=2.5](10,0,-10){A Spherical Mirror}%
\end{pspicture}}
```

pst-sigsys: Signal Processing



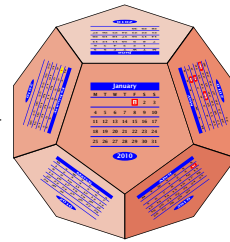
```
\usepackage{pstricks} \usepackage{pstadd}{pst-sigsys}
\begin{pspicture}(-2,-2)(5.5,2)
  \rput(-1.75,0){\node{xc}{x_c(t)}}
  \pscircleop[operation=times](0,0){otimes}
  \rput(0,1.25){\node{s}{s(t)}}
  \psblock[fillstyle=solid,fillcolor=purple!20]%
    (2.75,0){conv}{\parbox[c]{2\psunit}%
      {\centering Conversion from impulse
        train to discrete-time sequence}}
  \rput(5.5,0){\node{x}{x[n]}}
  \psset{style=Arrow}
  \ncline[nodesepA=.15]{xc}{otimes}
  \ncline[nodesepA=.15]{s}{otimes}
  \ncline{otimes}{conv}\naput{x_s(t)}
  \ncline[nodesepB=.15]{conv}{x}
  \psframe[linecolor=purple,linestyle=dashed,
    style=Dash](-.75,-1.5)(4.5,1.5)
  \rput(1.875,1.75){C/D Converter}
\end{pspicture}
```

pst-platon: Platonic solids



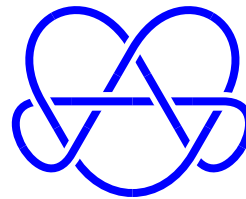
```
\usepackage{pst-platon}
\psTetrahedron\quad
\psDodecahedron
```

pst-calendar: Two or three dimensional calendars



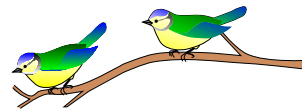
```
\usepackage{pst-calendar}
\begin{Example}[ltxps]{%
\psscalebox{0.13}{%
\psCalDodecaeder[Jahr=2010,
style=march]}}
```

pst-knot: Knot lines



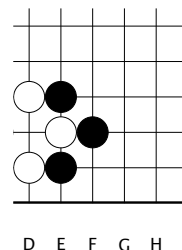
```
\usepackage{pst-knot}
\begin{pspicture}(-2,-2)(2,2)
\psKnot[linewidth=3pt,
linecolor=blue](0,0){7-7}
\end{pspicture}
```

pst-fun: "Funny objects"



```
\usepackage{pst-fun}
\begin{pspicture}(0,-1.3)(10,3)
\psBird[Branch]
\rput{-20}{(4,1.8){\psBird}
\end{pspicture}
```

psgo: Game "Go"



```
6 \usepackage{psgo}
5 \psscalebox{0.7}{%
4 \begin{psgopartialboard}[9]{{(4,1)(9,6)}
3 \stone{white}{c}{3} \stone{white}{e}{3}
2 \stone{white}{d}{2} \stone{white}{d}{4}
1 \stone{black}{f}{3} \stone{black}{e}{2}
\stone{black}{e}{4}
\end{psgopartialboard}}
```

1.2 PSTricks and PDF

Figure 2 on page 1014 shows the different ways generating a PDF output from the \TeX or \LaTeX source. With PSTricks only the greyed part can be used to generate a PDF. When not using the package `microtype` there will be no difference in the PDF, generated in the PSTricks way with `latex` or directly generated with `pdflatex`. When using a graphical user interface, e.g. Kile for Linux, `\TeX`shop for MAC OS, or `\TeX`niccenter for Windows, it is only one mouse click to generate the PDF output. The generated DVI or PostScript files have only a temporary character, they can be deleted after generating the PDF.

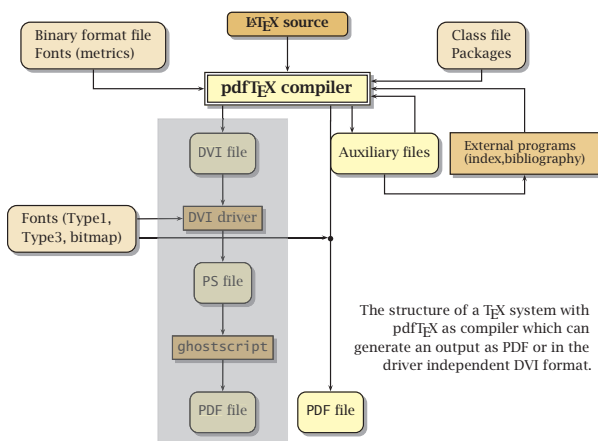


Figure 2: The default way for generating a PDF output with using PostScript code in the TeX source.

There are several ways of using PSTricks related code within a document which will be compiled with `pdflatex`, which allow to use other images in a PDF, PNG, or JPG format and, of course, the `microtype` package for an optimized text formatting.

1.3 pst2pdf

This is a Perl script which has to be used instead of the `pdflatex` command. The script extracts all `pspicture` or `postscript` environments from the main text body and runs these code snippets with the same preamble as the main document. The PDF output of these single documents then are cropped to get rid of the white space around the figure and also converted into the EPS and PNG format. The latter is only possible for running the script with Linux. After producing all PostScript related code as a single image, saved in a default directory `images/` the script `pst2pdf` runs the source in a last run with `pdflatex` and replaces all PostScript code with the former created image. This way of creating a PDF output is the best choice when all graphics as external images are needed.

The script has several optional arguments which are described with their defaults at the beginning of the script.

1.4 pst-pdf

This package from *Rolf Niepraschk* allows the cutting of the `pspicture` or `postscript` environments from the created DVI file into a new file `<file>-pics.ps`, which then is converted into a file `<file>-pics.pdf`. Every image will be on one page and the size of the image is taken from the `pspicture` coordinates or from the bounding box for a `postscript` environment. In a last `pdflatex` run the PDF images are inserted

instead of the PostScript related code. There are four steps needed:

1. `latex <file>`
2. `dvips -Ppdf -o <file>-pics.ps <file>.dvi`
3. `ps2pdf -dAutoRotatePages=/None <file>-pics.ps <file>-pics.pdf`
4. `pdflatex <file>`

Alternatively, one can use the script `ps4pdf` for these steps. The script is part of any TeX distribution, otherwise it is available on CTAN. There also exist some profiles for the graphical user interfaces. They are also available on CTAN (<http://mirror.ctan.org/graphics/pstricks/pst-support/>).

1.5 auto-pst-pdf

This package from *Will Robertson* works in the same way as `pst-pdf`, but it doesn't need a script or the four runs by the user. Everything is done in a `pdflatex` run, which is the reason why one has to enable the `shell-escape` option to allow the running from external programs from within `pdflatex`. MiKTeX users have to enable the option `enable-write18`. Special profiles are also available from CTAN (<http://mirror.ctan.org/graphics/pstricks/pst-support/>).

1.6 pdftricks

This package from *Radhakrishnan CV, Rajagopal CV, and Antoine Chambert-Loir* was the first one which supports PostScript related code with `pdflatex`. It works in a similar way as `pst-pdf`, but it needs some more code in the preamble to divide the PostScript part from the PDF part. More informations are available from CTAN or running `texdoc pdftricks`.

References

- [1] Frank Mittelbach, Michel Goossens, Sebastian Rahtz, Denis Roegel, and Herbert Voß. *The L^AT_EX Graphics Companion*. Addison-Wesley Publishing Company, Boston, second edition, 2006.
- [2] Herbert Voß. *PSTricks – Grafik für T_EX und L^AT_EX*. DANTE – Lehmanns, Heidelberg/Hamburg, fifth edition, 2008.

Markos Mitsos

Analyse von Kniffel und Ermittlung der optimalen Spielstrategie über APL

Das Spiel Kniffel ist zwar ein Glücksspiel, es enthält jedoch eine Entscheidung, welche der Spieler selbst treffen kann: Die Anzahl der Würfel, welche bei einem Zug zum Nachwurf verwendet werden. In diesem Artikel wird erklärt, wie dieses Spielelement verwendet werden kann, um, bedingte Wahrscheinlichkeiten nutzend, eine optimale Spielstrategie zu entwickeln.

Zu Beginn werden einige einfache Überlegungen zum Erstwurf angestellt, und gezeigt wie die Argumentation der elementaren Wahrscheinlichkeitslehre durch genauso elementare APL-Algorithmen ersetzt werden kann. Der Vorteil letzterer ist, dass sie auch beliebig komplexe Spielstellungen bewerten können. Eine solche Bewertung kann so viele Fallunterscheidungen benötigen, dass sie praktisch nicht mit Papier und Kugelschreiber zu bewältigen ist.

Die Bewertung beliebiger Würfe kann genutzt werden, um einen Erstwurf insgesamt, und zwar für jede beliebige, aber feste, Entscheidung zum Nachwurf, zu bewerten. Dies geschieht über bedingte Wahrscheinlichkeiten. Die optimale Strategie besteht darin, den (optimalen) Teilwurf zu wählen, welcher den bedingten Erwartungswert maximiert. Eine Strategie zur Spielstellung insgesamt ist optimal, falls sie es für jeden möglichen Erstwurf ist. So eine ist schon für sehr einfache Spielstellungen, z.B. „nur große Straße erwünscht“, wegen der vielen notwendigen Fallunterscheidungen, sehr schwer über elementare Argumente zu ermitteln.

Es werden Algorithmen vorgestellt, welche optimale Strategien für jede Spielstellung ermitteln. Die Ermittlung einer optimalen Spielstrategie erfordert eine sehr viel höhere Rechenleistung. Wegen gewisser Spielregeln, ist es dazu notwendig, alle möglichen Spielstellungen rückwärts zu analysieren. Eine kleine Oberfläche präsentiert dann das Ergebnis interaktiv.

Eine historische Bemerkung zum Arbeitsbereich ist angebracht: Er wurde entwickelt, um dabei APL+-Win zu erlernen und mit APL2 auf dem mainframe zu vergleichen. Insbesondere die erwähnte Oberfläche stellt ein Kennenlernen mit dem Konzept des WindowsInterface dar. Außerdem wurden die Algorithmen nur bis zur Funktionalität entwickelt und nie optimiert. Um parallel unter beiden Interpretern lauffähig zu sein, wurde unter Anderem großteils auf Kontrollstrukturen verzichtet. Eventuell wird in Zukunft eine deutlich verbesserte Version erstellt.

1 Analyse des Erstwurfs, Wahrscheinlichkeiten und Erwartungswerte verschiedener Ziele

In diesem Artikel wird das Spiel Kniffel analysiert. Bevor man sich an das Spiel als Ganzes macht, muss man einen Zug analysieren, davor jedoch erstmal den Erstwurf. Letzteres passiert im ersten Abschnitt.

1.1 Die Spielregeln von Kniffel und die verschiedenen Ziele

Da es Varianten von Kniffel mit leicht unterschiedlichen Regeln gibt, wird hier eine kurze Beschreibung der verwendeten gegeben. Das Spiel wird mit fünf (fairen, nicht unterscheidbaren) Würfeln gespielt. Es besteht aus 13 Zügen je Spieler. Der Ablauf eines Zuges ist folgender:

1. Der Spieler wirft alle 5 Würfel.
2. Der Spieler darf dann, nach freiem Ermessen, zwischen 0 und 5 Würfeln ein Mal nachwerfen.
3. Das Resultat wird mit dreizehn Zielen verglichen. In jeder Spielstellung sind einige dieser Ziele noch offen, andere sind erreicht oder gestrichen.
4. Das Resultat muss in einen offenen Posten eingetragen werden, oder es muss ein solcher gestrichen werden. Es darf jedoch nur eingetragen werden, falls das entsprechende Ziel erreicht worden ist.

Die Ziele und damit verbundenen Resultate sind, in der üblichen Reihenfolge:

1er bis 6er Beliebige Anzahl von Würfeln mit der vorgegebenen Augenzahl. Um konsistente Notation zu erreichen, betrachten wir das Ziel nur dann als erreicht, falls mindestens ein Würfel relevant ist. Die Summe der Augenzahlen der relevanten Würfel ergibt die Anzahl Punkte.

Dreier- und Viererpassch Mindestens 3 bzw. 4 Würfel mit gleicher Augenzahl. Die Summe aller Augenzahlen ergibt die Anzahl Punkte.

full house 3 Würfel mit gleicher Augenzahl und 2 weitere mit untereinander gleicher, aber von den drei ersten verschiedener, Augenzahl. 25 Punkte.

Kleine und große Straße 4 bzw. 5 Würfel mit ununterbrochen streng aufsteigender Augenzahl. 30 bzw. 40 Punkte.

Kniffel 5 Würfel mit gleicher Augenzahl. 50 Punkte.

Chance Beliebige Würfel. Die Summe aller Augenzahlen ergibt die Anzahl Punkte.

Im Rahmen des Spieles gibt es zwei weitere Regeln zur Bewertung von Zielen, welche später, in Zusammenhang mit der Spielstrategie und auf Seite 19, erläutert werden. Bevor man überhaupt definieren kann, was eine solche ist, muss man sich etwas mit der reinen Glücksspielseite eines Erstwurfs auseinandersetzen. Dann wird bereits klar, wozu entsprechende Algorithmen nützlich sind, welche das Problem automatisch lösen.

1.2 Wahrscheinlichkeiten verschiedene Ziele über den Erstwurf zu erreichen

In Zusammenhang mit der Analyse der reinen Glücksspielseite eines Erstwurfs sprechen wir von einem Ergebnis oder einer Menge $\{x_1, x_2, x_3, x_4, x_5\}$ und meinen damit, dass wir die geworfenen Würfel in diese Reihenfolge bringen können. Meistens verlangen wir $x_i \leq x_j$, falls $i < j$. Unter dieser Voraussetzung sind Ergebnisse eindeutig. Wir tun jedoch auch gegebenenfalls so, als ob die Würfel nummeriert sind. Dann sprechen wir vom Tupel $(x_1, x_2, x_3, x_4, x_5)$ und meinen damit, dass der erste Würfel x_1 zeigt, der zweite x_2 etc.

An einigen Stellen werden wir die Menge

$$\mathcal{W} = \{\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}\}$$

aller möglicher Augenzahlen eines Würfels brauchen. Um die Notation klar von jener eines Wurf-Ergebnisses zu trennen, werden wir dafür stets \mathcal{W} verwenden. Da man mit 5 Würfeln würfelt, ist ein Tupel Element des kartesischen Produktes, $(x_1, x_2, x_3, x_4, x_5) \in \mathcal{W}^5$, während eine Menge eine Äquivalenzklasse ist,

$$\{x_1, x_2, x_3, x_4, x_5\} \in \mathcal{W}^5 / S_5.$$

Es gibt insgesamt

$$6^5 = 7.776$$

(als Tupel) verschiedene Erstwürfe. Diese sind natürlich nicht alle (als Mengen) unterschiedlich. Das Ergebnis $\{\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}\}$ kann nur dann zu Stande kommen, wenn alle Würfel 1 zeigen, entspricht also nur dem Tupel $(\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix})$. Das Ergebnis $\{\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}\}$ hingegen entspricht vielen Tupeln, z.B. $(\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix})$ oder $(\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix})$. Das elementare Argument aus der Oberstufe lautet, dass man die $\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}$ an 5 Stellen legen kann, die $\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}$ dann an eine der restlichen 4 etc., so dass man insgesamt

$$\binom{5}{1} = 5! = 5 \times 4 \times 3 \times 2 \times 1 = 120 \quad (1)$$

Möglichkeiten hat, das Ergebnis zu erzeugen. (Der erste benutzte Ausdruck wird „5 über 1“, der zweite „5 Fakultät“ ausgesprochen.) Ähnlich gehören zu der Menge $\{\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}\}$ genau

$$\binom{5}{2} = \frac{5!}{2! \times 3!} = \frac{120}{2 \times 6} = 10 \quad (2)$$

Tupel.

Diese elementare Überlegungen genügen schon, um die Wahrscheinlichkeit zu ermitteln, bestimmte Ziele mit einen einzigen (Erst-) Wurf zu erreichen. Hat man z.B. als einziges Ziel Kniffel zu würfeln, so beachtet man, dass es 6 verschiedene Kniffel gibt, nämlich

$$\{\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \dots, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}\} \text{ und } \{\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \dots, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{smallmatrix}\}.$$

Damit ist die Erfolgswahrscheinlichkeit

$$6 \times 1 = 6$$

$$W(\text{Kniffel}) = W((1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 0, 1)) = \frac{6 \times 1}{7.776} = \frac{1}{1.296} \approx 0.08\%. \quad (3)$$

Wahrscheinlichkeiten werden meistens in Prozent, auf 2 Nachkommastellen gerundet, angegeben.

Die Schreibweise in der Klammer soll bedeuten, dass von den dreizehn möglichen Zielen zwölf, durch 1 charakterisiert, bereits erreicht sind, und eins, nämlich das zwölfte, Kniffel, noch offen ist. Die Ziele sind in der zu Beginn angegebenen Reihenfolge aufgelistet, „obere“ sind von „unteren“ mit einem Semikolon getrennt. Die Notation werden wir bei Bedarf leicht erweitern, und -1 für ein gestrichenes Ziel benutzen und/oder die erreichte Anzahl Punkte anstelle von 1 für erreichte Ziele eintragen.

Genauso gibt es zwei große Straßen, nämlich

$$\{\square, \square, \square, \square, \square\} \text{ und } \{\square, \square, \square, \square, \square\},$$

und es gilt, mit (1),

$$W(\text{große Straße}) = W((1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 0, 1, 1)) = \frac{2 \times 120}{7.776} = \frac{5}{162} \approx 3.09\%, \quad (4)$$

während es $6 \times 5 = 30$ verschiedene full house gibt, nämlich

$$\{\square, \square, \square, \square, \square\}, \{\square, \square, \square, \square, \square\}, \dots \{\square, \square, \square, \square, \square\} \text{ und } \{\square, \square, \square, \square, \square\},$$

und damit, sowie (2),

$$W(\text{full house}) = W((1, 1, 1, 1, 1, 1 ; 1, 1, 0, 1, 1, 1, 1)) = \frac{30 \times 10}{7.776} = \frac{25}{648} \approx 3.86\% \quad (5)$$

gilt.

Etwas anders ist die Situation bei 1er. Hier ergibt sich die Erfolgswahrscheinlichkeit am einfachsten als Komplement des Misserfolges, also der Wahrscheinlichkeit, dass kein Würfel 1 zeigt:

$$\begin{aligned} W(1er) &= W((0, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1, 1)) = 1 - W(\text{keine } \square) \\ &= 1 - \left(\frac{5}{6}\right)^5 = 1 - \frac{3.125}{7.776} = \frac{4.651}{7.776} \approx 59.81\%. \end{aligned} \quad (6)$$

Man kann sich mit etwas mehr Mühe eine Tabelle mit den Wahrscheinlichkeiten zum Erreichen jedes der dreizehn einzelner Ziele zusammenstellen. Allerdings führen diese Methoden nicht sehr weit, aus zwei Gründen.

Zum einen will man die Erfolgswahrscheinlichkeit, also das Erreichen eines beliebigen, noch offenen Ziel, für jede beliebige Spielstellung bestimmen. Dies bedeutet die Erstellung einer Tabelle mit $2^{13} = 8.192$ Einträgen – in Wirklichkeit, und wegen der Zusatzregeln aus 4.2, sogar deutlich mehr. Die meisten dieser Einträge werden auch nur mit sehr viel komplexeren Argumenten und Fallunterscheidungen zu bestimmen sein.

Nur in wenigen Fällen kann man die Wahrscheinlichkeiten addieren, einzelne Ziele zu erreichen, um die Wahrscheinlichkeit zu erhalten, ein beliebiges dieser zu erzielen. Bei den drei zuerst untersuchten Zielen, Kniffel, große Straße und full house, ist dies der Fall, da es keinen Erstwurf gibt, welcher mehr als ein Ziel erfüllt. Dies ist jedoch die Ausnahme. Für 1er, Dreierpasch und Kniffel z.B. trifft es definitiv nicht zu.

Zum anderen sind die Wahrscheinlichkeiten, ein Ziel zu Erreichen, eigentlich nicht das, was für das Spiel wichtig ist! Man ist an möglichst vielen Punkten interessiert, was die Situation komplett ändert, wie im nächsten Abschnitt erklärt.

1.3 Erwartungswerte verschiedener Ziele bei Durchführung des Erstwurfs

Hat man als einziges Ziel Kniffel zu erreichen, ist es relativ egal, ob man von Wahrscheinlichkeiten oder Punkten spricht. Der Grund ist, dass man exakt 50 Punkte für einen Erfolg bekommt, sonst nichts. Im Allgemeinen ist das nicht so einfach. In der Wahrscheinlichkeitslehre spricht man von Erwartungswert und definiert ihn als die gewichtete Summe (der Punkte) aller Ergebnisse. Es gilt also, wegen (3),

$$\begin{aligned} E(\text{Kniffel}) &= E((1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 0, 1)) \\ &= \left(50 \times \frac{1}{1.296}\right) + \left(0 \times \frac{1.295}{1.296}\right) = \frac{25}{648} \approx 0.0386 \text{ (Punkte)}. \end{aligned} \quad (7)$$

Erwartungswerte werden wir meistens in Punkten, ohne die „Einheit“, auf 4 Nachkommastellen gerundet, angeben.

Genauso ist, wegen (4),

$$E(\text{große Straße}) = E((1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 0, 1, 1)) = 40 \times \frac{5}{162} = \frac{100}{81} \approx 1.2346, \quad (8)$$

und, wegen (5),

$$E(\text{full house}) = E((1, 1, 1, 1, 1, 1 ; 1, 1, 0, 1, 1, 1, 1)) = 25 \times \frac{25}{648} = \frac{625}{648} \approx 0.9645. \quad (9)$$

Aber schon der Erwartungswert der 1er ist nicht so einfach zu ermitteln. Ohne ins Detail zu gehen, muss man dazu die Wahrscheinlichkeit ermitteln, eine bestimmte Anzahl \square exakt zu würfeln, wobei noch diverse Unterfälle zu unterscheiden sind, und kann dann den Erwartungswert als Summe ermitteln. Die manuelle Aufstellung einer Tabelle mit den dreizehn Erwartungswerten zu einzelnen Zielen ist eine nicht unerhebliche Fleißaufgabe.

Bei der gleichzeitigen Betrachtung mehrerer Ziele werden die Probleme deutlich komplexer. Man muss sehr viele Fälle unterscheiden, da z.B. bei der Betrachtung der Ziele Dreierpasch und full house, und für Ergebnisse wie $\{\square, \square, \square, \square, \square, \square\}$ oder $\{\square, \square, \square, \square, \square, \square\}$, stets abgewogen werden muss, als was diese zu werten sind, um die maximale Anzahl Punkte zu erreichen. (Dass wir dieses Ziel verfolgen und auch dass wir stets in der Lage sind zu erkennen, welches Resultat das beste ist, nehmen wir natürlich an!)

Letztendlich wird man bei vielen Kombinationen von Zielen nicht mit elementaren Argumenten weiterkommen. Man wird große Listen anfertigen müssen, welche alle 7.776 mögliche Erstwürfe enthalten und anhand dieser alle Fälle klassifizieren und bewerten müssen. Das schreit jedoch nach maschineller Unterstützung.

1.4 Maschinelle Bewertung des Erstwurfs und Ermittlung aller Wahrscheinlichkeiten und Erwartungswerte

Um sowohl die Wahrscheinlichkeit bestimmte, beliebig vorgegebene Ziele, zu erreichen, wie auch den zugehörigen Erwartungswert zu berechnen, implementieren wir einen einfachen Algorithmus in APL, genauer code welcher unter APL2 und APL+Win lauffähig ist. Die in diesem Unterabschnitt folgenden Befehle sind Teil der Funktion KN_WURF_BEWERTEN, welche im Anhang nochmals vollständig abgebildet wird. Der Algorithmus benötigt den Vektor *HABE* als Parameter, der angibt welche Ziele noch offen sind, und *WS*, einen binären Skalar, der gesetzt wird, wenn Wahrscheinlichkeiten (und nicht Erwartungswerte) gesucht werden. Der Parameter *GEWICHT* wird in 2.3 erklärt, und ist momentan ein Vektor der nur Einsen enthält, *SPIEL* steht momentan auf 0 und ist zu ignorieren. Zu ignorieren ist auch *FOLGE*, sowie das vierzehnte Element von *HABE*.

Der Algorithmus funktioniert für einen beliebigen (Erst-) *WURF*, genauer für ein beliebiges array von Würfeln, dessen letzte Dimension die Länge 5 hat und die gewürfelten Augenzahlen darstellt. In vielen Fällen benutzen wir für den Wert von Vektoren der Länge 13, z.B. *HABE*, die Darstellung

$$HABE = (x_1, x_2, x_3, x_4, x_5, x_6 ; x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}),$$

welche an die natürliche Unterteilung der Kniffel-Ziele erinnern soll. Insbesondere charakterisieren wir die Stellung innerhalb eines Spieles stets durch so einen Vektor, welchen wir auch stets *HABE* nennen und als APL-Variable behandeln.

Ein interessantes array, in diesem Fall eine Matrix, von Würfeln erhalten wir z.B., indem wir mittels

```
EW_UNG←⊖1+(5ρ6)τ~1+ι65
```

alle 7.776 mögliche Erstwürfe erzeugen und als *WURF* verwenden. Die Bezeichnung *EW_UNG* reservieren wir für diese Menge aller ungewichteter Erstwürfe, welche wir mit W^5 gleichsetzen (siehe aber auch 2.3).

Als erstes wird in KN_WURF_BEWERTEN (über einen der ältesten APL-Befehle überhaupt) in *TREFFER*, eine Matrix der Breite 6, notiert wie oft jede Augenzahl geworfen wurde:

```
[21] TREFFER←+/[ρρWURF]WURF∘.(ι6) ρ GETROFFENE AUGENZAHLN
```

Effektiv werden dadurch Permutationen ignoriert, und man unterscheidet nicht Tupel sondern Mengen von Würfeln. Im nächsten Schritt wird in *WERT* die Anzahl Punkte notiert die jeder Wurf bringen würde,

jedoch ohne Rücksicht auf Erfolg und Offenheit des Zieles. Dann wird in *TREFFER* beschlossen, welche Ziele tatsächlich erreicht wurden. Somit kann nun in *WERT* entschlossen werden, welche Punkte je Ziel zu holen sind (z.B. (5, 0, 0, 0, 0, 0 ; 0, 0, 0, 0, 0, 50, 5) für den Wurf $\{\square, \square, \square, \square, \square, \square\}$, falls 1er, Kniffel und Chance offen, Dreier- und Viererpassch jedoch bereits erreicht sind):

```
[30]  ⌊ 'WERT'←', ((WS)/'1' ), '(V+(35×SPIEL×(63>+/0[HABE[16])×(63≤(+/0[HABE[16]))+V)
      )+(100×SPIEL×(50=HABE[12])×(5=TREFFER)))', ((+/WURF), (+/WURF), (((-1+ρWURF),
      4)ρ25 30 40 50), (+/WURF))+(7/[ρρTREFFER], [0.1+1+ρρTREFFER]100×SPIEL×(50=H
      ABE[12])×(v/(5=TREFFER))))' ⌈ BEWERTUNG DER VERSCHIEDENEN ZIELE
[31]  TREFFER←(0≠TREFFER), (v/(TREFFER∈3 4 5)), (v/(TREFFER∈4 5)), (((v/(2=TREFFER
      ))^ (v/(3=TREFFER))), (v/((4ρ1)⊆(0≠TREFFER))), [0.1+1+ρρTREFFER] (v/((5ρ1)⊆(0
      ≠TREFFER))))v (3/[ρρTREFFER], [0.1+1+ρρTREFFER] (SPIEL^((50=HABE[12])^ (v/(5=T
      REFFER))))), (v/(5=TREFFER)), [0.1+1+ρρTREFFER]1 ⌈ TREFFER: BESTIMMTE AUGEN
      /DREIERPASS/VIERERPASS/FULL HOUSE/KLEINE STRASSE/GROSSE STRASSE/KNIFFEL/CHA
      NCE
[32]  WERT←((0=HABE)×[ρρTREFFER]TREFFER×WERT), 0 ⌈ WURF BEWERTEN
```

Nun wählt der code (natürlich) stets für jedes Tupel das erreichte Ziel mit der maximalen Punktzahl, bei Gleichheit das erste. (Im Beispiel also Kniffel, falls dieser jedoch schon erreicht ist 1er.) Genauso wählt er bei Wahrscheinlichkeiten für jedes Tupel das erste erreichte Ziel. Insbesondere stimmen die zwei für die möglichen Werte von *WS* gewählten Ziele nicht unbedingt für jedes Tupel überein. Die Auswahl wird genutzt, um ungewollte Resultate zu verwerfen:

```
[34]  ⌊ 'TREFFER'←(1=+\TREFFER['', ((-1+ρρTREFFER)/';'), '8 7, (16), (8+16)])['', ((-1+ρρ
      TREFFER)/';'), '(2+16), 2 1, (8+16)]^ (TREFFER←(WERT+FOLGE)=14/[ρρWERT], [0.1+
      1+ρρWERT] / (WERT+FOLGE))' ⌈ MEHRFACH-TREFFER
[35]  WERT←TREFFER×WERT ⌈ VERWERFEN
```

Im letzten Schritt werden Erfolge längs der vorletzten Achse gewichtet summiert. Im Fall des erzeugten *EW_UNG* heißt das einfach „alle Ergebnisse addieren“:

```
[38]  WERT←(0≠V)×(+/[1+ρρWERT] (14/[1+ρρGEWICHT], [0.1+ρρGEWICHT]GEWICHT)×WERT)÷(
      V←14/[ρρGEWICHT], [0.1+1+ρρGEWICHT]+/[ρρGEWICHT]GEWICHT) ⌈ WERTE GEWICHTET
      UND NORMIERT SUMMIEREN
```

Nun kann man natürlich trivialerweise alle einzelnen Ziele untersuchen. Es ergibt sich sofort, und natürlich in Übereinstimmung mit den bereits manuell gewonnenen Resultate, die Tabelle 1.

Tabelle 1: Erfolgswahrscheinlichkeit und Erwartungswert aller möglicher Erstwürfe für einzelne Kniffel-Ziele

Ziel	Erfolgswahrscheinlichkeit [%]	Erwartungswert [Punkte]
1er	59.81	0.8333
2er	59.81	1.6667
3er	59.81	2.5000
4er	59.81	3.3333
5er	59.81	4.1667
6er	59.81	5.0000
Dreierpasch	21.30	3.7269
Viererpasch	2.01	0.3511
full house	3.86	0.9645
kleine Straße	15.43	4.6296
große Straße	3.09	1.2346
Kniffel	0.08	0.0386
Chance	100.00	17.5000

Im Unterschied zur manuellen Bestimmung, ist jedoch jede Kombination von Zielen nicht schwieriger als Einzelziele. Wir gehen als Beispiel in Tabelle 2 die Kombination der bereits einzeln untersuchten vier Ziele 1er, full house, große Straße und Kniffel an.

Tabelle 2: Erfolgswahrscheinlichkeit und Erwartungswert aller möglicher Erstwürfe für eine Kombination von vier Kniffel-Zielen

$HABE = (0, 1, 1, 1, 1, 1 ; 1, 1, 0, 1, 0, 0, 1)$		
Ziel	Erfolgswahrscheinlichkeit [%]	Erwartungswert [Punkte]
Summe	63.99	3.0228
1er	59.81	0.7851
full house	2.57	0.9645
große Straße	1.54	1.2346
Kniffel	0.06	0.0386

Man beachte, dass, wie erwartet, die 1er dem Kniffel zwar Wahrscheinlichkeiten, jedoch keine Punkte (im Vergleich zu den Resultaten zu Einzelzielen) abnehmen können. Dies liegt daran, dass jeder Wurf, der beide Ziele erfüllt, zwar mehr Punkte als Kniffel bringt, jedoch in beiden Fällen exakt einen „Wahrscheinlichkeits-Erfolg“ – und dann greift die Reihenfolge der Ziele.

2 Optimales Verhalten bei einem konkreten Erstwurf

Bis jetzt hatten wir keine Chance irgendetwas zu entscheiden, bis auf die Wahl des erreichten Zieles in manchen Fällen. Wir haben es als selbstverständlich angesehen, dass dabei stets die maximale Punkt- oder allgemeiner Erfolgszahl gewählt wird. Damit lässt die Analyse des Erstwurfs keine Spielräume zu: Die Erwartungswerte und die Erfolgswahrscheinlichkeiten sind dieselben für jeden Spieler. Eine Strategie kommt erst durch den Nachwurf ins Spiel.

2.1 Bedingte Wahrscheinlichkeiten und das optimale Verhalten bei einem konkreten Erstwurf und dem Ziel Kniffel

Wir nehmen nun an, dass wir z.B. $\{\square, \square, \square, \square, \square\}$ gewürfelt haben, und als einziges Ziel Kniffel haben. Die nahe liegenden Fragen sind: Ist es wichtig welche Würfel wir erneut werfen? Durch welche Wahl maximieren wir unsere Erfolgschancen? In diesem ganz einfachen Fall können wir mit einigen Argumenten weiterkommen. Wir brauchen den Begriff der bedingten Wahrscheinlichkeit. Wir schreiben

$$W(\text{Kniffel} \mid \{\square, \square, \square\}) = W((1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 0, 0, 1) \mid \{\square, \square, \square\})$$

für die Wahrscheinlichkeit Kniffel unter der Bedingung zu würfeln, dass wir den Teilwurf $\{\square, \square, \square\}$ vom Erstwurf behalten. Allgemein schreiben wir

$$W(HABE \mid TW) \quad \text{und} \quad E(HABE \mid TW)$$

entsprechend für die bedingte Wahrscheinlichkeit und den bedingten Erwartungswert die in $HABE$ noch offenen Ziele unter der Voraussetzung zu erhalten, dass wir den Teilwurf TW behalten.

Konkret ist es bei Kniffel als (einzigem) Ziel klar, dass

$$W((1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 0, 0, 1) \mid TW) = E((1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 0, 0, 1) \mid TW) = 0$$

für alle Teilwürfe TW gilt, welche mindestens zwei verschiedene Elemente enthalten (da sich solche nicht zu Kniffel ergänzen lassen). Außerdem kennen wir (aus den Gleichungen (3) und (7))

$$W((1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 0, 0, 1) \mid \emptyset) \approx 0.08 \quad \text{und} \quad E((1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 0, 0, 1) \mid \emptyset) \approx 0.0386.$$

Man beachte, dass es durchaus legitim ist, keinen Würfel zu behalten.

Ansonsten ist es egal, welche Augenzahlen wir auszeichnen und behalten, im Folgenden Sinne: Es ist sicher gleichgültig ob man zwei \square ($TW = \{\square, \square\}$) behält und versucht drei weitere \square zu würfeln, oder

zwei $\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}$, $\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}$, $\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}$, $\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}$ oder $\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}$ als Teilwurf auswählt. Wie hoch ist die Erfolgswahrscheinlichkeit? Es ist in allen Fällen vorgegeben, was jeder der Würfel des Nachwurfs zeigen muss, und man erhält die bedingten Wahrscheinlichkeiten nach derselben Überlegungen wie bei der Ermittlung von $E(\text{Kniffel})$ selbst.

Tabelle 3: Bedingte Wahrscheinlichkeiten und Erwartungswerte für das Ziel Kniffel und die verschiedenen möglichen Teilwürfe

Teilwurf TW	Erfolgswahrscheinlichkeit [%]	Erwartungswert [Punkte]
\emptyset	0.08	0.0386
$x \in \mathcal{W}$		
$\{x\}$	0.08	0.0386
$\{x, x\}$	0.46	0.2315
$\{x, x, x\}$	2.78	1.3889
$\{x, x, x, x\}$	16.67	8.3333
$\{x, x, x, x, x\}$	100.00	50.0000
sonst	0.00	0.0000

Für den Erstwurf $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ hätten wir also die besten Erfolgschancen, falls wir $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ oder $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ behalten würden. Dies hätten wohl auch fast alle Spieler intuitiv getan.

2.2 Optimaler Teilwurf für konkrete Erstwürfe und die Grenzen der manuellen Bearbeitung

Wir definieren nun den bezüglich Wahrscheinlichkeiten optimierten Teilwurf, oder W-optimalen Teilwurf, $OTW = OTW^W(HABE; EW)$, für jede Teilmenge von Zielen $HABE$ und jeden Erstwurf EW , als den Teilwurf $OTW \subset EW$ für den

$$W(HABE; EW) := W(HABE | OTW) \geq W(HABE | TW), \forall TW \subset EW \quad (10)$$

gilt. Analog definieren wir den E-optimalen Teilwurf OTW^E als den Teilwurf $OTW \subset EW$ mit

$$E(HABE; EW) := E(HABE | OTW) \geq E(HABE | TW), \forall TW \subset EW. \quad (11)$$

Die so definierten, ausgezeichneten Wahrscheinlichkeiten und Erwartungswerte sind die maximal erreichbaren. Nach dem eben über Kniffel Gesagten gilt z.B.

$$\begin{aligned} OTW^W((1, 1, 1, 1, 1, 1; 1, 1, 1, 1, 1, 0, 1); \{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}) = \\ = OTW^E((1, 1, 1, 1, 1, 1; 1, 1, 1, 1, 1, 0, 1); \{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}) = \{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}. \end{aligned}$$

Man beachte, dass ein optimaler Teilwurf nicht eindeutig sein muss. Dies ist insbesondere im Beispiel der Fall, auch $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ ist optimal. Die nächste Frage ist natürlich, wie schwer es ist, optimale Teilwürfe zu bestimmen - beim Ziel Kniffel war ja einfach.

Wir versuchen, den optimalen Teilwurf zum selben Erstwurf und für das Ziel große Straße zu bestimmen. Hier ist die Situation gewissermaßen komplementär zu Kniffel, es gilt

$$W((1, 1, 1, 1, 1, 1; 1, 1, 1, 1, 0, 1, 1) | TW) = E((1, 1, 1, 1, 1, 1; 1, 1, 1, 1, 0, 1, 1) | TW) = 0$$

für alle Teilwürfe, welche mindestens zwei gleiche Elemente und/oder $\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}$ sowie $\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}$ enthalten (da letzteres für keine große Straße zutrifft). Außerdem kennen wir (aus den Gleichungen (4) und (8))

$$W((1, 1, 1, 1, 1, 1; 1, 1, 1, 1, 0, 1, 1) | \emptyset) \approx 3.09 \quad \text{und} \quad E((1, 1, 1, 1, 1, 1; 1, 1, 1, 1, 0, 1, 1) | \emptyset) \approx 1.2346.$$

Versucht man jedoch, alle nicht leeren Teilmengen von $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ und $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ zu untersuchen (eben alle noch nicht erwähnten möglichen Teilwürfe!), so sieht man, dass man sehr viele Fallunterscheidungen machen muss. Für den Teilwurf $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ z.B. führt nur der Nachwurf $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ zum Ziel, während für $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ sowohl $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ wie auch $\{\begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}, \begin{smallmatrix} \blacksquare & \bullet \\ \bullet & \bullet \end{smallmatrix}\}$ passend sind.

Man sieht zum Glück relativ leicht, dass man die Fälle, wie in diesen zwei Beispielen, folgendermaßen klassifizieren kann: Man muss Teilmengen bestimmter Kardinalität unterscheiden, und diese wiederum jeweils in solche, welche $\{\square\}$ oder $\{\square, \blacksquare\}$ enthalten, und damit nur eine Vervollständigung erlauben, und die restlichen, welche zwei Vervollständigungen zulassen, unterteilen. Die Wahrscheinlichkeit, eine solche Vervollständigung zu erzielen, erhält man über ähnliche Argumente, wie die zur Bestimmung von W (große Straße).

Tabelle 4: Bedingte Wahrscheinlichkeiten und Erwartungswerte für das Ziel große Straße und die verschiedenen möglichen Teilwürfe

Teilwurf TW	Erfolgswahrscheinlichkeit [%]	Erwartungswert [Punkte]
\emptyset	3.09	1.2346
$x_i \in \mathcal{W}$ $x_i < x_j$ falls $i < j$ $ \text{TW} \cap \{\square, \blacksquare\} \leq 1$		
$\{x_1\}, \text{TW} \cap \{\square, \blacksquare\} \neq \emptyset$	1.85	0.7407
$\{x_1\}, \text{TW} \cap \{\square, \blacksquare\} = \emptyset$	3.70	1.4815
$\{x_1, x_2\}, \text{TW} \cap \{\square, \blacksquare\} \neq \emptyset$	2.78	1.1111
$\{x_1, x_2\}, \text{TW} \cap \{\square, \blacksquare\} = \emptyset$	5.56	2.2222
$\{x_1, x_2, x_3\}, \text{TW} \cap \{\square, \blacksquare\} \neq \emptyset$	5.56	2.2222
$\{x_1, x_2, x_3\}, \text{TW} \cap \{\square, \blacksquare\} = \emptyset$	11.11	4.4444
$\{x_1, x_2, x_3, x_4\}, \text{TW} \cap \{\square, \blacksquare\} \neq \emptyset$	16.67	6.6667
$\{x_1, x_2, x_3, x_4\}, \text{TW} \cap \{\square, \blacksquare\} = \emptyset$	33.33	13.3333
$\{x_1, x_2, x_3, x_4, x_5\}$	100.00	40.0000
sonst	0.00	0.0000

Insbesondere gilt

$$\begin{aligned} OTW^W((1, 1, 1, 1, 1, 1; 1, 1, 1, 1, 0, 1, 1); \{\square, \square, \square, \square, \blacksquare\}) = \\ = OTW^E((1, 1, 1, 1, 1, 1; 1, 1, 1, 1, 0, 1, 1); \{\square, \square, \square, \square, \blacksquare\}) = \{\square, \blacksquare\}. \end{aligned}$$

Auch dieser optimale Teilwurf ist nicht eindeutig. Hier treten auch schon intuitiv nicht unmittelbar einleuchtende Relationen auf, z.B. dass es für große Straße schlechter ist $\{\square\}$ zu haben als nichts, und dass $\{\square, \blacksquare\}$ besser ist als $\{\square, \square\}$, aber genau so gut wie $\{\square, \square, \blacksquare\}$.

Ähnlich lange Listen und komplexere Überlegungen sind für die Bestimmung der optimalen Teilwürfe zum Erstwurf $\{\square, \square, \square, \square, \blacksquare\}$ und für die restlichen einfachen Ziele zu erwarten. Außerdem ist vollkommen unklar, ob z.B.

$$OTW^W(\text{Dreierpasch}; \{\square, \square, \square, \square, \blacksquare\}) = OTW^E(\text{Dreierpasch}; \{\square, \square, \square, \square, \blacksquare\})$$

gilt. Das Hauptproblem, und damit eine erste Grenze der manuellen Bearbeitung, wird jedoch durch Vergleich von Kniffel und großer Straße klar. Wir haben nicht nur komplett unterschiedliche optimale Teilwürfe ermittelt, sondern auch nicht vereinbare Strategien verwendet, um die mögliche Teilwürfe zu klassifizieren und somit die Probleme überhaupt von Hand angehen zu können.

2.3 Zusammenfassung und Gewichtung von gleichen Erstwürfen und Teilwürfen

Wir wollen nun versuchen, optimale Teilwürfe mittels APL zu ermitteln. Im Prinzip wollen wir dazu für einen konkreten Erstwurf alle möglichen Teilwürfe auflisten, und für jeden dieser alle möglichen Nachwürfe bewerten. Dabei werden natürlich sehr viele gleichwertige Konstellationen bearbeitet. Um das Verfahren durchsichtiger zu gestalten, ist ein technischer Schritt angebracht. Bei der Betrachtung allgemeinerer Probleme in den nächsten Abschnitten ist er sogar aus Speicherplatz- und Laufzeit-Gründen notwendig. Es handelt sich um die Gewichtung von Würfeln.

Eine Gewichtung hätte auch schon bei der Bewertung des Erstwurfs verwendet werden können. Wir hatten 7.776 Erstwürfe untersucht, welche jedoch nur als Tupel unterscheidbar sind. Als Menge sind natürlich viele davon gleich, z.B. $(\square, \square, \square, \square, \square)$, $(\square, \square, \square, \blacksquare, \square)$, $(\square, \square, \blacksquare, \square, \square)$, $(\square, \blacksquare, \square, \square, \square)$ und $(\blacksquare, \square, \square, \square, \square)$. Der in 1.4 vorgestellte Bewertungs-Algorithmus wandelt sowieso alle fünf in die Zeile

410000 von *TREFFER* um. Es ist also ohne Weiteres möglich (aufsteigend sortierte) Mengen und nicht Tupel zu bewerten und gleichzeitig zu gewichten. Im Beispiel würde man (nur) die Menge $\{\square, \square, \square, \square, \square\}$ mit dem Gewicht 5 in die Bewertung schicken.

Die Gewichtung übernimmt die Funktion *KN_WURF_GEWICHTEN*. Sie funktioniert mit beliebigen arrays *WURF*, z.B. Matrizen. Es wird nicht angenommen, dass die letzte Dimension die Länge 5 hat. Letzteres ist beim Erstwurf der Fall, es sollen jedoch auch Teilwürfe gewichtet werden können. Im ersten Teil werden Augenzahlen gezählt:

```
[0]  ERG←KN_WURF_GEWICHTEN WURF;TREFFER;GEWICHT;I;II;III
[1]  A
[2]  A SORTIERT UND GEWICHTET EINE ODER MEHRERE LISTEN VON KNIFFEL-WUERFEN
[3]  A
[4]  kn_gui←⊆∈('kn_gui' '4=⊆NC ''⊆AV''')[1+2≠⊆NC 'kn_gui'] A STANDARD-UMGEBUNG
    IN ABHAENGIGKEIT VOM AV
[5]  I←ρWURF A STRUKTUR NOTIEREN
[6]  WURF←,[ι~1+ρI]WURF A ALLE WUERFE ZU EINER LISTE ZUSAMMENFASSEN
[7]  TREFFER←+/[2]WURF∘.=(ι6) A WUERFEL NACH AUGENZAHL GRUPPIEREN
```

Im zweiten Teil wird jedes Tupel in die aufsteigende Sortierung gebracht, welche wir meistens für Mengen verwendet haben. Diese setzt sich auch natürlich zu einer Sortierung der Mengen fort:

```
[8]  WURF←(ρWURF)ρ(,WURF)[⊆((~1↑ρWURF)/ι×/~1↓I),((~1↑I)-,TREFFER)[((~1↑I)/6×~1+
    ι×/~1↓I)+,WURF],[1.1](,WURF)] A JEDEN WURF GRUPPIEREN UND SORTIEREN
[9]  WURF←WURF[III←⊆(II←I[~1+ρI]/ι×/~2↓I),WURF;] A WUERFE JEDER LISTE
[10] TREFFER←TREFFER[III;] A UND ZUGEOERIGE TREFFER NATUERLICH ANORDNEN
```

Es gibt jedoch auch eine weitere Art eine Menge von Würfeln anzuordnen, nämlich den größten aus Würfeln mit gleicher Augenzahl bestehenden Teilwurf nach vorne zu ziehen. Dass diese gewissermaßen natürlich ist, haben wir bereits gesehen: Wir haben sie zur Herleitung der Aufstellungen 3 und 4 verwendet. Diese Sortierung wird nun erzwungen, und es werden Tupel zu Mengen zusammengefasst:

```
[11] WURF←II,WURF[⊆(⊆II),(:,⌈/TREFFER),[1.1](,+(2=TREFFER))];] A WUERFE JEDER LI
    STE NACH DEREN STRUKTUR ANORDNEN, LISTEN-ZUGEOERIGKEIT NOTIEREN
[12] II←1,1↓((⊆/(WURF≠~1⊆[1]WURF))/ι1↑ρWURF) A GLEICHARTIGE WUERFE GRUPPIEREN,
[13] WURF←WURF[II;] A BEIBEHALTEN
[14] GEWICHT←(∧/0≠WURF)×((1↓II,(1+×/~1↓I))-II) A UND GEWICHTEN
```

Der letzte Teil befasst sich mit arrays höherer Dimension. Deren Bedeutung wird im nächsten Abschnitt erklärt:

```
[15] II←WURF[;1]ιι⌈/WURF[;1] A ZU JEDER LISTE GHOERENDE WUERFE ZAEHLEN
[16] II←(III∘.≥(ι⌈/III))×(II∘.+(~1+ι⌈/(III←(1↓II,(1+ρWURF[;1]))-II))) A UND IND
    IZIEREN
[17] WURF←((~2↓I),⌈/III),(~1↑I))ρ(0,[1]WURF)[1+,II;1+ι~1↑I] A WUERFE
[18] GEWICHT←((~2↓I),⌈/III))ρ(0,GEWICHT)[1+,II] A UND GEWICHTUNGEN IN DIE URSP
    RUENGLICHEN LISTEN UNTERTEILEN
[19] ERG←WURF GEWICHT A ERGEBNIS UEBERGEHEN
```

Die Anwendung der Funktion auf *EW_UNG* liefert eine deutlich kürzere Liste *EW_GEW* von 252 Erstwürfen und einen Vektor *GEW_EW* mit 252 Gewichten. Ähnlich wie in 1.4, reservieren wir die Bezeichnungen für den Quotientenraum $EW_GEW = \mathcal{W}^5 / S_5$ und die Mächtigkeiten der Äquivalenzklassen *GEW_EW*. (Selbstverständlich liefert *KN_WURF_BEWERTEN* für diese Parameter dasselbe wie für *EW_UNG*!) Die Gewichtung verallgemeinert die in (1) bis (5) verwendeten Anzahlen und Gewichte zur Tabelle 5, welche im nächsten Abschnitt, z.B. zur Erstellung von 8, wichtig sein wird. Man kann die Tabelle natürlich genau so gut über kombinatorische Argumente manuell erstellen.

Tabelle 5: Unterteilung aller möglicher Erstwürfe in Typen und die zugehörigen Gewichte

Typ	Struktur	Beispiel	Mengen	jeweiliges Gewicht	Tupel
Summe			252		7.776
$x_i \in \mathcal{W}$ $x_i \neq x_j$ falls $i \neq j$					
$\{x_1, x_1, x_1, x_1, x_1\}$	5		6	1	6
$\{x_1, x_1, x_1, x_1, x_2\}$	4		30	5	150
$\{x_1, x_1, x_1, x_2, x_2\}$	3+2		30	10	300
$\{x_1, x_1, x_1, x_2, x_3\}$	3		60	20	1.200
$\{x_1, x_1, x_2, x_2, x_3\}$	2+2		60	30	1.800
$\{x_1, x_1, x_2, x_3, x_4\}$	2		60	60	3.600
$\{x_1, x_2, x_3, x_4, x_5\}$	0		6	120	720

2.4 Maschinelle Ermittlung der optimalen Teilwürfe für konkrete Erstwürfe

Nun sind wir in der Lage, die maschinelle Ermittlung der optimalen Teilwürfe $OTW^W(HABE; EW)$ und $OTW^E(HABE; EW)$ für beliebige, aber feste, Kombinationen von Zielen und Erstwürfen durchzuführen. Der folgende Code ist Teil von KN_ERSTWURF_BEWERTEN. Diese Funktion verwendet als Unterfunktionen KN_WURF_BEWERTEN und KN_WURF_GEWICHTEN. Sie kann mit arrays arbeiten, wird hier jedoch mit einem Vektor der Länge 5 als $ERSTWURF$ aufgerufen.

Die Bewertung geht einfach alle Möglichkeiten durch, 0, 1, 2, 3, 4 oder 5 Würfel (von jedem angegebenen Erstwurf) zu behalten, und zwar in einer klassischen APL2-Schleife, welche sich als inhaltlich sinnvoll und aus Speicherplatzgründen notwendig erweist:

```
[10] I←¯1↓ρERSTWURF
[11] TEILWURF←(I,0)ρ<0ρ0◇WERT←(I,0,(13+SPIEL))ρ0
...
[16] I←0
[17] L01:
...
[29] I←I+1
[30] →(I≤5)/L01
```

Zu jeder dieser Anzahlen gibt es unterschiedlich viele Möglichkeiten den Teilwurf als Tupel zu bestimmen. Nicht alle führen zu verschiedenen Mengen. (Behalten wir von die drei Würfel , so ist es egal welche und wir nachwerfen.) Deshalb ist es sinnvoll, die Teilwürfe zu sortieren und zu gewichten:

```
[18] ⍥'WURF←((¯1↓ρERSTWURF),(1 5 10 10 5 1)[1+I],I)ρERSTWURF[',((¯1+ρρERSTWURF)
/';'),(ε('0ρ0' '15' '1 2 1 3 1 4 1 5 2 3 2 4 2 5 3 4 3 5 4 5' '1 2 3 1 2 4
1 2 5 1 3 4 1 3 5 1 4 5 2 3 4 2 3 5 2 4 5 3 4 5' '1 2 3 4 1 2 3 5 1 2 4 5
1 3 4 5 2 3 4 5' '15')[1+I]),')' ⍥ I WUERFEL BEHALTEN
[19] WURF←1▷(KN_WURF_GEWICHTEN WURF) ⍥ DIE ZUGEOERIGEN TEILWUERFE ANORDNEN,
[20] TEILWURF←TEILWURF,[ρρTEILWURF](ε[ρρWURF]WURF) ⍥ NOTIEREN
```

Jeder der so erhaltenen Teilwürfe muss durch alle möglichen Nachwürfe ergänzt werden, was eine dritte Auffächerung der Möglichkeiten ergibt. Auch hier ist eine Gewichtung angemessen, wobei der Rang der Objekte größer 2 ist:

```
[22] WURF←,[0.1+¯1+ρρWURF]WURF ⍥ UND
[23] ⍥'WURF←',(ε(5-¯1↑ρWURF)/ε'((6/[¯1+ρρWURF])'),'WURF',(ε(ε'),[ρρWURF]((¯2+ρW
URF),(6'),''(⍥'15-¯1↑ρWURF)''(ε'))ρ16))' ⍥ ERGAENZEN
[24] (WURF GEWICHT)←KN_WURF_GEWICHTEN WURF ⍥ WUERFE GEWICHTEN
```

Im Hauptteil wird ein Quader von Würfeln bewertet. Er enthält alle Nachwürfe zu allen Teilwürfen zu einer bestimmten Anzahl zu behaltender Würfel. Die Ergebnisse, also die Werte der untersuchten Teilwürfe, werden über die Schleife in *WERTE* akkumuliert:

```
[27] →(0≡(ERG←(HABE GEWICHT WS SPIEL)KN_WURF_BEWERTEN WURF))/0 A GEWICHTET UND
    NORMIERT BEWERTEN
[28] WERT←WERT,[←1+ρρWERT]ERG A TEILRESULTAT ANHAENGEN
```

Schließlich wird (außerhalb der Schleife), der optimale Teilwurf bestimmt, indem die erste Zeile aus *WERTE* mit dem optimalen Resultat ausgewählt wird:

```
[31] I←(1=+I)∧(I←V=(←1↑ρV)/[ρρV],[0.1+←1+ρρV]⌈/(V←+/WERT)) A (ERSTE) OPTIMALE
    STRATEGIE FUER JEDEN ERSTWURF ERMITTELN
[32] ⌊'TEILWURF←',((1=ρρI)/'ρ'),'←(←1↓ρI)ρ(,I)/,TEILWURF' A JEWEILS BEIZUBEHALTE
    NDE WUERFEL ANGEBEN
[33] ⌊'WERT←((←1↓ρI),(13+SPIEL))ρ(,I)⌈,[←1+ρρWERT]WERT' A ZUGEOERIGE ERWARTUN
    GSWERTE ANGEBEN
[34] ERG←TEILWURF WERT A ERGEBNISSE UEBERGEBEN
```

Damit können wir nun maschinell optimale Teilwürfe und die zugehörigen Wahrscheinlichkeiten bzw. Erwartungswerte für beliebige, aber feste, Erstwürfe leicht ermitteln. Man beachte, dass man die Resultate von 1 fast durchgehend verbessert.

Tabelle 6: Aufstellung der optimalen Teilwürfe für einzelne Kniffel-Ziele und den Erstwurf $EW = \{\cdot, \cdot, \cdot, \cdot, \cdot\}$

HABE je nach Ziel und $EW = \{\cdot, \cdot, \cdot, \cdot, \cdot\}$				
Ziel	$OTW^W(.; EW)$	$W(.; EW)$ [%]	$OTW^E(.; EW)$	$E(.; EW)$ [Punkte]
1er	$\{\cdot\}$	100.00	$\{\cdot, \cdot\}$	2.5000
2er	$\{\cdot, \cdot\}$	100.00	$\{\cdot, \cdot, \cdot\}$	5.0000
3er	\emptyset	59.81	\emptyset	2.5000
4er	\emptyset	59.81	\emptyset	3.3333
5er	$\{\cdot, \cdot, \cdot\}$	100.00	$\{\cdot, \cdot, \cdot\}$	8.3333
6er	\emptyset	59.81	\emptyset	5.0000
Dreierpasch	$\{\cdot, \cdot, \cdot\}$	44.44	$\{\cdot, \cdot, \cdot\}$	5.9444
Viererpasch	$\{\cdot, \cdot, \cdot, \cdot\}$	7.41	$\{\cdot, \cdot, \cdot, \cdot\}$	0.8657
full house	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	33.33	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	8.3333
kleine Straße	\emptyset	15.43	\emptyset	4.6296
große Straße	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	5.56	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	2.2222
Kniffel	$\{\cdot, \cdot, \cdot\}$	0.46	$\{\cdot, \cdot, \cdot\}$	0.2315
Chance	\emptyset	100.00	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	19.0000

Genauso funktioniert natürlich auch die Bestimmung optimaler Teilwürfe zu konkreten Erstwürfen und Spielstellungen mit mehreren Zielen, z.B. die bereits in 2 untersuchte benutzte Kombination von vier Zielen.

Tabelle 7: Untersuchung der optimalen Teilwürfe für eine Kombination von vier Kniffel-Zielen und den Erstwurf $\{\cdot, \cdot, \cdot, \cdot, \cdot\}$

HABE = (0, 1, 1, 1, 1, 1 ; 1, 1, 0, 1, 0, 0, 1) und $EW = \{\cdot, \cdot, \cdot, \cdot, \cdot\}$			
	$OTW^W(HABE; EW) = \{\cdot\}$	$OTW^E(HABE; EW) = \{\cdot, \cdot, \cdot, \cdot, \cdot\}$	
Ziel	$W(HABE; EW)$ [%]	$E(HABE; EW)$ [Punkte]	
Summe	100.00	9.6667	
1er	100.00	1.3333	
full house	0.00	8.3333	
große Straße	0.00	0.0000	
Kniffel	0.00	0.0000	

Die Verfolgung der zwei teuren, aber harten, Ziele lohnt sich in diesem Fall nicht, sie werden fallen gelassen. Hier verbessert man das Resultat von 2 deutlich.

3 Die optimale Strategien zu einer Stellung: Optimales Verhalten bei jedem Erstwurf

Wir haben bereits gesehen, wie wir, mit Hilfe von Argumenten oder durch maschinelles Nachrechnen, unser Verhalten für jede vorgegebene Kombination von Zielen und jeden konkreten Erstwurf optimieren können. Dabei ist es wesentlich, ob wir versuchen die Wahrscheinlichkeit zu optimieren eins der Ziele zu erreichen, oder unsere Punktzahl. Der nächste Schritt ist quasi zwingend.

3.1 Definition der optimalen Strategien einer Stellung bezüglich Wahrscheinlichkeiten und Erwartungswerten

Das Optimum aus einer (Spiel-) Stellung, also einer vorgegebenen Kombination von Zielen, holen wir sicherlich dadurch raus, dass wir für jeden Erstwurf den optimalen Teilwurf bestimmen und wählen.

Wir definieren also die bezüglich Wahrscheinlichkeiten optimale (Stellungs-) Strategie, oder W-optimale Strategie als die Menge

$$OS^W(\text{HABE}) := \{OTW^W(\text{HABE}; \text{EW}); \text{EW} \in \text{EW_UNG} = \mathcal{W}^5\}. \quad (12)$$

Analog bilden wir die bezüglich Erwartungswerten optimale (Stellungs-) Strategie, oder E-optimale Strategie

$$OS^E(\text{HABE}) := \{OTW^E(\text{HABE}; \text{EW}); \text{EW} \in \text{EW_UNG} = \mathcal{W}^5\}. \quad (13)$$

Es genügt natürlich die optimalen Teilwürfe für die 252 gewichtete Erstwürfe aus EW_GEW zu bestimmen, es müssen nicht alle 7.776 mögliche Erstwürfe aus EW_UNG untersucht werden.

Passend zu den optimalen Strategien definieren wir noch die (maximale) Stellungen-Wahrscheinlichkeit

$$\begin{aligned} W_{OS}(\text{HABE}) &:= \frac{\sum_{\text{EW} \in \text{EW_UNG} = \mathcal{W}^5} W(\text{HABE}; \text{EW})}{|\mathcal{W}|^5} \\ &= \frac{\sum_{\text{EW} \in \text{EW_GEW}} \text{GEW_EW}(\text{EW}) \times W(\text{HABE}; \text{EW})}{|\mathcal{W}|^5} \end{aligned} \quad (14)$$

und den (maximalen) Stellungen-Erwartungswert

$$\begin{aligned} E_{OS}(\text{HABE}) &:= \frac{\sum_{\text{EW} \in \text{EW_UNG} = \mathcal{W}^5} E(\text{HABE}; \text{EW})}{|\mathcal{W}|^5} \\ &= \frac{\sum_{\text{EW} \in \text{EW_GEW}} \text{GEW_EW}(\text{EW}) \times E(\text{HABE}; \text{EW})}{|\mathcal{W}|^5}. \end{aligned} \quad (15)$$

Die jeweils erste Summe läuft über alle möglichen Erstwürfe, wobei für jeden Erstwurf EW definitionsgemäss (nach (10) bzw. (11)) der entsprechende optimale Teilwurf $OTW^{\cdot}(\text{HABE}; \text{EW})$ verwendet wird. Die jeweils zweite Summe läuft über die gewichteten Erstwürfe und die Gewichtung ist zwar mathematisch nicht ganz sauber hingeschrieben, APLmäÙig jedoch klar!

3.2 Optimale Strategien zum Einzelziel Kniffel und die (endgültige) Grenze der manuellen Bearbeitung

Nun wurde das Problem formalisiert, und wir können versuchen $OS^W(\text{Kniffel}) = OS^E(\text{Kniffel})$, sowie $W_{OS}(\text{Kniffel})$ und $E_{OS}(\text{Kniffel})$ zu bestimmen. Die Tabelle 3 enthält alle notwendigen Informationen um die beiden optimalen Strategien zu bestimmen (und zu sehen, dass sie übereinstimmen): Man muss stets die Würfel mit gleicher Augenzahl behalten, welche die größte solche Gruppe bilden. Dies ist auch intuitiv einleuchtend, allerdings nicht ganz einfach präziser zu formulieren, ohne auf eine vollständige Liste von Fällen zurückzugreifen.

Einen Weg bietet eine Unterteilung der gewichteten Erstwürfe aus 5, welche auch für die Bestimmung der zwei gewünschten Zahlen, $W_{OS}(\text{Kniffel})$ und $E_{OS}(\text{Kniffel})$, notwendig ist. Kombiniert man sie mit der, bereits erwähnten, Tabelle 3, so erhält man die Aufstellung 8

Tabelle 8: Bestimmung der Stellungen-Wahrscheinlichkeit und - Erwartungswertes zum Ziel Kniffel anhand der Klassifikation der Erstwürfe nach Struktur

Struktur EW gleiche Würfel	Anzahl	Einzel- Gewicht	Gesamt- Gewicht	$W(\text{Kniffel}; \cdot)$ [%]	$E(\text{Kniffel}; \cdot)$ [Punkte]
(gewichtete) Summe	252		7.776	1.26	0.6316
5	6	1	6	100.00	50.0000
4	30	5	150	16.67	8.3333
3+2	30	10	300	2.78	1.3889
3	60	20	1.200	2.78	1.3889
2+2	60	30	1.800	0.46	0.2315
2	60	60	3.600	0.46	0.2315
1	6	120	720	0.08	0.0386

$W_{OS}(\text{Kniffel})$ ist damit immerhin mehr als 15 Mal höher als die entsprechende Wahrscheinlichkeit Kniffel direkt zu würfeln ($W(\text{Kniffel}) \approx 0.08$), was natürlich auch für das Verhältnis von $E_{OS}(\text{Kniffel})$ zum Erwartungswert $E(\text{Kniffel}) \approx 0.0386$ gilt.

Noch schneller als bei dem bereits untersuchten Problem der Bestimmung optimaler Teilwürfe, stößt man bei der manuellen Ermittlung optimaler Strategien an Grenzen. Schon für Kniffel war die präzise Niederschrift von OS nicht trivial, und die Berechnung von W_{OS} und E_{OS} hat einige Vorarbeiten in Anspruch genommen.

Das nächste Beispiel wäre die große Straße. Auch hier haben wir bereits die Tabelle 4 bereitgestellt, welche es erlaubt, die optimale Strategie mehr oder minder zu beschreiben. Während bei Kniffel jedoch „die größte Gruppe behalten“ den Sachverhalt gut in Worte fasste, kann man hier nicht „möglichst viele Unterschiedliche behalten“ verwenden, da man z.B. den Erstwurf $\{\square, \square, \square, \square, \square\}$ am besten komplett verwirft, und von $\{\square, \square, \square, \square, \square\}$ können sicher nicht alle Würfel gebraucht werden.

Versucht man nun die Stellungswerte des Zieles große Straße zu berechnen, so werden die Probleme deutlich größer. Will man mit den gewichteten Erstwürfen, wie eben, arbeiten, so kommt man mit der benutzten Unterteilung nicht weiter, da man z.B. $\{\square, \square, \square, \square, \square\}$ verwerfen, von $\{\square, \square, \square, \square, \square\}$ jedoch $\{\square\}$ behalten muss. Man kann aber auch nicht einfach anders sortieren, da die Fälle aus den (sieben) Klassen in 4 auch unterschiedliche Gewichte haben. Man muss also die Liste weiter in Unterklassen auffächern - sicher mit Kugelschreiber und Papier machbar, aber sicher nicht angenehm.

Die Betrachtung von nur zwei Zielen, z.B. Kniffel und große Straße, gleichzeitig, führt jedoch zu einer solchen Anzahl von zu betrachtenden Fällen, dass es sehr wohl denkbar ist, dass durch manuelle Bearbeitung daraus ein Lebenswerk wird.

3.3 Maschinelle Ermittlung optimaler Strategien für beliebige, aber feste, Stellungen

Eine maschinelle Lösung ist natürlich möglich. Bevor wir einen Teil von `KN_STELLUNG_BEWERTEN` vorstellen, sind an dieser Stelle ein paar Überlegungen zu Laufzeiten und Speicherplatz angebracht. Der einfachste Weg eine Stellung komplett zu bewerten, ist es sicher für alle 7.776 Erstwürfe jeweils alle Teilwürfe als Option zu betrachten und alle deren Nachwürfe zu bewerten. Wie viele sind das? Da ein Erstwurf 5 Elemente hat,

und wir für jedes bestimmen können, ob es zum Teilwurf gehören soll, gibt es $2^5 = 32$ Teilwürfe. (Anders ausgedrückt ist der TW ein Element der Potenzmenge $\mathcal{P}(EW)$ des Erstwurfs.) Es ergibt sich die Aufstellung 9.

Tabelle 9: Klassifikation der Teilwürfe eines Erstwurfs nach Kardinalität und die Anzahl resultierenden Nachwürfe

$ TW $	Anzahl	$ \{NW\} $ (einzeln)	Nachwürfe insgesamt
Summe	32		16.807
0	1	7.776	7.776
1	5	1.296	6.480
2	10	216	2.160
3	10	36	360
4	5	6	30
5	1	1	1

Man muss also insgesamt $7.776 \times 16.807 = 130.691.232$ Nachwürfe untersuchen. Dass sind Zahlen mit denen man leicht den Speicherplatz sprengt, und auch die Laufzeit wäre so sicher schlecht. Eine Vereinfachung hatten wir schon eingeführt, wir hatten gleichwertige Erstwürfe gewichtet und zusammengefasst. Damit beginnt auch die Bewertung einer Stellung:

```
[13] ⍺←'ERSTWURF←',(⊖5/⊖'((6≠'),'(1 0ρ0)',(⊖(⊖'),((6'),''(⊖''⊖5),''(⊖')ρ⊖6)))'⍺
    ALLE MOEGLICHEN ERSTWUERFE
[14] (ERSTWURF GEWICHT)←KN_WURF_GEWICHTEN ERSTWURF ⍺ MOEGLICHE ERSTWUERFE GEWIC
    HTEN
[15] ERSTWURF←ERSTWURF[I←⍺GEWICHT;]⊖GEWICHT←GEWICHT[I] ⍺ AUFSTEIGEND NACH GEWIC
    HT SORTIEREN (EIGENTLICH UNNOETIG)
[16] TEILWURF←0ρ⊖0ρ0⊖WERT←(0,(13+SPIEL))ρ0
```

Eine zweite Vereinfachung ermöglicht die Klassifikation 9 in Zusammenhang mit 5. Sie erlaubt es, Klassen von Erstwürfen zu bilden, welche sich unter KN_ERSTWURF_BEWERTEN und den zwei in dieser Funktion vorkommenden Gewichtungen identisch (was Anzahl und Struktur von Nachwürfen anbelangt) verhalten werden. Dies berücksichtigt die Hauptschleife der Funktion, welche sieben Mal durchlaufen wird:

```
[20] V←((⊖ρGEWICHT)=GEWICHT⊖GEWICHT)/GEWICHT ⍺ GRUPPEN VON ERSTWUERFEN NACH GEW
    ICHT ZUSAMMENFASSEN
...
[24] I←1
[25] L01:
...
[29] I←I+1
[30] →(I≤ρV)/L01
```

Der Hauptteil ist trivial, und besteht nur aus dem Aufruf von KN_ERSTWURF_BEWERTEN und der Sammlung der Ergebnisse:

```
[26] →(0≡(ERG←HABE WS SPIEL I KN_ERSTWURF_BEWERTEN(V[I]=GEWICHT)≠ERSTWURF))/0 ⍺
    JEWEILS EINE GRUPPE (NORMIERT) BEWERTEN
[27] TEILWURF←TEILWURF,(1>ERG)⊖WERT←WERT,[1](2>ERG) ⍺ GEWICHTUNG NOCH NICHT BER
    EUCKSICHTIGEN
```

Damit ist die optimale Strategie bestimmt (eine Zuordnung vor Erstwürfen zu optimalen Teilwürfen). Um die Stellungen-Wahrscheinlichkeit bzw. den -Erwartungswert zu bestimmen benötigt man noch eine Gewichtung:

```
[35] WERT←(GEWICHT×[1]WERT)÷(+/GEWICHT) ⍺ WERTE GEWICHTEN UND NORMIEREN
```

Spätestens an dieser Stelle sollte auch klar werden, warum alle bis jetzt vorgestellten Funktionen mit höherdimensionalen arrays arbeiten können müssen: Sie ergeben sich auf natürliche Art und Weise bei der Bildung aller Nachwürfe. Genauer werden $7 \times 6 = 42$ (Strukturklassen Mal Anzahl zu behaltender Würfel) arrays vom Rang 4 für jede Stellung bewertet. Insgesamt sind dies 254.100 Würfe, also weniger als 0.2% von obiger, naiver Schätzung. Eine Zusammenfassung und Vermeidung der Schleifen wäre möglich, dann müssten jedoch 6.350.400 Würfe, inklusive sehr vieler Platzhalter, bewertet werden.

JEWELLS EINE GRUPPE (NORMIERT) BEWERTEN
 [27] $TEILWURF \leftarrow TEILWURF, (1 \triangleright ERG) \diamond WERT \leftarrow WERT, [1] (2 \triangleright ERG) \text{ A } GEWICHTUNG \text{ NOCH NICHT BER}$
 EUCKSICHTIGEN

Damit ist die optimale Strategie bestimmt (eine Zuordnung von Erstwürfen zu optimalen Teilwürfen). Um die Stellungen-Wahrscheinlichkeit bzw. den -Erwartungswert zu bestimmen benötigt man noch eine Gewichtung:

[35] $WERT \leftarrow (GEWICHT \times [1] WERT) \div (+/GEWICHT) \text{ A } WERTE \text{ GEWICHTEN UND NORMIEREN}$

Spätestens an dieser Stelle sollte auch klar werden, warum alle bis jetzt vorgestellten Funktionen mit höherdimensionalen arrays arbeiten können müssen: Sie ergeben sich auf natürliche Art und Weise bei der Bildung aller Nachwürfe. Genauer werden $7 \times 6 = 42$ (Strukturklassen Mal Anzahl zu behaltender Würfel) arrays vom Rang 4 für jede Stellung bewertet. Insgesamt sind dies 254.100 Würfe, also weniger als 0.2% von obiger, naiver Schätzung. Eine Zusammenfassung und Vermeidung der Schleifen wäre möglich, dann müssten jedoch 6.350.400 Würfe, inklusive sehr vieler Platzhalter, bewertet werden.

3.4 Beispiele (maximaler) Stellungen-Wahrscheinlichkeiten und -Erwartungswerte sowie optimaler Strategien

Damit können nun trivialerweise die (maximalen) Stellungswerte für die alle einzelnen Ziele ermittelt werden. Es ergibt sich die Tabelle 10, von der wir nur einen Wert auch manuell ermittelt haben. Man vergleiche auch mit der Aufstellung 1.

Tabelle 10: Stellungen-Wahrscheinlichkeiten und -Erwartungswerte für alle einzelnen Kniffel-Ziele

HABE je nach Ziel		
Ziel	$W_{OS}(\cdot)$ [%]	$E_{OS}(\cdot)$ [Punkte]
1er	83.85	1.5278
2er	83.85	3.0556
3er	83.85	4.5833
4er	83.85	6.1111
5er	83.85	7.6389
6er	83.85	9.1667
Dreierpasch	54.13	10.2087
Viererpasch	13.23	2.4215
full house	19.12	4.7806
kleine Straße	41.18	12.3536
große Straße	12.97	5.1899
Kniffel	1.26	0.6316
Chance	100.00	21.2500

Auch hier funktioniert die Analyse natürlich auch für jede Stellung mit mehreren Zielen, z.B. die bereits mehrmals benutzte Kombination von vier Zielen, genauso. Die Ergebnisse sind natürlich besser als die in 2.

Tabelle 11: Stellungen-Wahrscheinlichkeit und -Erwartungswert für einer Kombination von vier Kniffel-Zielen und deren Verteilung auf letztere

$HABE = (0, 1, 1, 1, 1, 1; 1, 1, 0, 1, 0, 0, 1)$		
Ziel	$W_{OS}(HABE)$ [%]	$E_{OS}(HABE)$ [Punkte]
Summe	87.03	8.7085
1er	81.35	0.7353
full house	3.50	3.5365
große Straße	2.10	4.0123
Kniffel	0.09	0.4244

An dieser Stelle werden vermutlich die optimalen Strategien stark voneinander differieren. Wir geben einen Auszug aus jeder, genauer den optimalen Teilwurf für den ersten und den letzten Erstwurf aus jeder der sieben Klassen aus der Tabelle 5.

Tabelle 12: Auszug aus der optimalen Strategie für Wahrscheinlichkeiten (in Prozent) und für eine Kombination von vier Kniffel-Zielen

HABE = (0, 1, 1, 1, 1, 1 ; 1, 1, 0, 1, 0, 0, 1)							
Erstwurf	Gewicht	$OTW_W(\text{HABE}; \cdot)$	$W(\text{HABE}; \cdot)$	davon entfallen auf die Ziele			
				1er	full house	große Straße	Kniffel
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	1	$\{\cdot\}$	100.00	100.00	0.00	0.00	0.00
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	1	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	100.00	0.00	0.00	0.00	100.00
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	5	$\{\cdot\}$	100.00	100.00	0.00	0.00	0.00
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	5	\emptyset	63.99	59.81	2.57	1.54	0.06
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	10	$\{\cdot\}$	100.00	100.00	0.00	0.00	0.00
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	10	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	100.00	0.00	100.00	0.00	0.00
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	20	$\{\cdot\}$	100.00	100.00	0.00	0.00	0.00
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	20	\emptyset	63.99	59.81	2.57	1.54	0.06
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	30	$\{\cdot\}$	100.00	100.00	0.00	0.00	0.00
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	30	\emptyset	63.99	59.81	2.57	1.54	0.06
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	60	$\{\cdot\}$	100.00	100.00	0.00	0.00	0.00
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	60	\emptyset	63.99	59.81	2.57	1.54	0.06
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	120	$\{\cdot\}$	100.00	100.00	0.00	0.00	0.00
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	120	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	100.00	0.00	0.00	100.00	0.00

Man beachte, dass die „davon“ Wahrscheinlichkeiten nicht die bedingten Wahrscheinlichkeiten der einzelnen Ziele sind, da beim Nachwurf mehrere Ziele erreicht werden können und nur eins davon zählt. Das gilt auch für die nächste Tabelle.

Tabelle 13: Auszug aus der optimalen Strategie für Erwartungswerte (in Punkten) und für eine Kombination von vier Kniffel-Zielen

HABE = (0, 1, 1, 1, 1, 1 ; 1, 1, 0, 1, 0, 0, 1)							
Erstwurf	Gewicht	$OTW_E(\text{HABE}; \cdot)$	$E(\text{HABE}; \cdot)$	davon entfallen auf die Ziele			
				1er	full house	große Straße	Kniffel
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	1	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	50.0000	0.0000	0.0000	0.0000	50.0000
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	1	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	50.0000	0.0000	0.0000	0.0000	50.0000
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	5	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	11.6667	3.3333	0.0000	0.0000	8.3333
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	5	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	8.5000	0.1667	0.0000	0.0000	8.3333
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	10	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	25.0000	0.0000	25.0000	0.0000	0.0000
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	10	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	25.0000	0.0000	25.0000	0.0000	0.0000
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	20	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	7.6389	2.7778	3.4722	0.0000	1.3889
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	20	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	5.1389	0.2778	3.4722	0.0000	1.3889
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	30	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	9.6667	1.3333	8.3333	0.0000	0.0000
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	30	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	8.5000	0.1667	8.3333	0.0000	0.0000
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	60	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	7.6667	1.0000	0.0000	6.6667	0.0000
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	60	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	6.8333	0.1667	0.0000	6.6667	0.0000
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	120	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	40.0000	0.0000	0.0000	40.0000	0.0000
$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	120	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	40.0000	0.0000	0.0000	40.0000	0.0000

Die beiden Strategien sind, wie erwartet, komplett unterschiedlich. Es werden verschiedene Ziele verfolgt und Erfolge anders auf Ziele verteilt.

4 Die optimalen Strategien des gesamten Spieles: Optimales Verhalten bei allen Stellungen und jedem Erstwurf

Jetzt könnte man denken, wir sein fertig mit der Analyse von Kniffel. Eigentlich kann man eine Oberfläche bereitstellen, und zu jeder angegebenen Stellung auf Knopfdruck die optimale W- und E-Strategie bestimmen und anzeigen. Leider ist das Ganze nicht so einfach.

4.1 Komplikationen durch „leichte“ und „harte“ Ziele und die Betrachtung der Folge-Stellungen jedes Spielzuges

Das erste, prinzipielle Problem ist, dass es „leichte“ und „harte“ Ziele gibt. Jeder weiß natürlich intuitiv, dass full house leichter zu erreichen ist als Kniffel. Dies haben wir ja auch mit Wahrscheinlichkeiten belegt. Allerdings bleibt die Frage: Um wie viel leichter? Das Verhältnis der Punkte (25 : 50) ist sicher kein guter Anhaltspunkt, wie die Auflistungen 1 und 10 zeigen – Kniffel scheint so hart zu sein, dass es sich nicht lohnt danach zu streben. Deshalb auch die Anführungszeichen, wir können die „Härte“ eines Ziels noch nicht präzise fassen. Wir haben zwar bereits gezeigt, was jedes Ziel in einem Zug, für sich betrachtet, wert ist, nicht jedoch wie schwer es im Rahmen des Spieles zu erreichen ist.

Dies hat natürlich Folgen für die optimale Strategie. Als einfachstes Beispiel betrachte man den Erstwurf $\{\square, \square, \square, \square, \blacksquare\}$ und die Ziele Viererpasch und Kniffel. Eine W-optimale Strategie für die Stellung ist es sicher nichts zu tun, ein Ziel ist bereits (mit 100% Wahrscheinlichkeit) erreicht. Wir *wissen* aber, dass dies im Rahmen des Spieles ein gravierender Fehler ist, man *muss* einfach die \blacksquare nachwerfen. Noch schlimmer: Sogar beim Erstwurf $\{\square, \square, \square, \square, \square\}$ ist es für dieselbe Stellung und wegen der Reihenfolge der Ziele W-optimal das Ziel Viererpasch einzutragen!

Ein echtes Beispiel mit Erwartungswerten ist nicht trivial zu konstruieren. Sucht man jedoch z.B. gezielt die Unterschiede in den Strategien zu vier Zielen, welche in der später in diesem Abschnitt erscheinenden Tabelle 18 bewertet werden, so findet man abweichende Entscheidungen. Z.B. ist es für den Erstwurf $\{\square, \square, \square, \square, \blacksquare\}$ und die Stellung an sich optimal $\{\square, \square, \square\}$ zu behalten. Damit behält man sich die Ziele 1er und große Straße offen. Hat man jedoch 59 Punkte im oberen Block, so sind die so maximal zu erreichenden 3 \square im Rahmen des Spiels ein schlechtes Geschäft. Deshalb ist dann $\{\square, \square\}$ optimal, also höhere Wahrscheinlichkeit für große Straße, auf die Gefahr hin, ein Ziel streichen zu müssen, und mit der Chance in einem der drei nächsten Züge 4 \square zu erreichen (siehe dazu auch 4.2).

Die einzige Methode die Ziele rigoros zu unterteilen, ist die Betrachtung aller Folge-Stellungen. Kann man jeder solchen einen Wert zuordnen, so kann man letzteren in die Bewertung, welche bei der Bestimmung der optimalen Strategie einer Stellung vorgenommen wird, einfließen lassen, und somit eine optimale Spiel-Strategie der Stellung erhalten. Da jedoch (fast) jede Folge-Stellung ihrerseits solche hat, sind wir gezwungen das Spiel insgesamt zu betrachten.

Das Spiel beginnt stets mit $HABE = (0, 0, 0, 0, 0, 0 ; 0, 0, 0, 0, 0, 0)$. Bei jedem Zug verschwindet eine Null, bis man bei $HABE = (1, 1, 1, 1, 1, 1 ; 1, 1, 1, 1, 1, 1)$ landet. Man beachte nochmals, dass wir, zumindest bis jetzt, nicht zwischen erreichten und gestrichenen Zielen unterscheiden. Das Spiel hat $2^{13} = 8.192$ Stellungen. Wie viele davon nach dem Zug x , $x \in \{0, 1, \dots, 13\}$ erreichbar sind, gibt die binomische Formel an.

Tabelle 14: Anzahl Spiel-Stellungen nach Anzahl erreichter oder gestrichener Ziele, ohne Unterscheidung letzterer Fälle sowie ohne Zusatzregeln

0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	13	78	286	715	1.287	1.716	1.716	1.287	715	286	78	13	1

(Die Anzahl möglicher Spiele, nur nach Reihenfolge des Erreichens/Streichens der Ziele unterteilt, ist hingegen $13! = 6.227.020.800$). Genau so wie für das Spiel insgesamt, gibt die binomische Formel auch für jede konkrete Spiel-Stellung an, wie viel Folge-Stellungen beliebiger Generation sich aus dieser ergeben. Für

acht bereits erreichte Ziele z.B., führt der nächste Zug zu einer von 5 möglichen Folge-Stellungen, insgesamt sind es $1 + 5 + 10 + 10 + 5 + 1 = 32$, inklusive der zu untersuchenden.

Nach dem eben Erklärten, muss man zur Analyse einer Spiel-Stellung alle Folge-Stellungen bereits kennen, und dies muss zwangsweise vom Ende ausgehend rekursiv erfolgen. Man könnte natürlich jedes Mal neu anfangen, es ist jedoch sicher besser einen Vektor mit 8.192 Elementen anzulegen. Diesen kann man mittels decode/encode mit den verschiedenen Werten von *HABE* assoziieren, und in einem längeren Lauf befüllen. Dann wird man in Zukunft stets alle nötigen Folge-Stellungen für jede vorgegebene Spiel-Stellung zur Verfügung haben, und wird KN_STELLUNG_BEWERTEN nutzen können, um alles über diese Spiel-Stellung zu erfahren. (Die vollständigen Strategien für jede Spiel-Stellung zu speichern ist wahrscheinlich aus Speicherplatzgründen unpraktisch.) Leider ist auch das noch nicht Alles.

4.2 Komplikationen durch zusätzliche Spielregeln und Abhängigkeit der Bewertung eines Wurfs von der Spiel-Stellung

Ein weiteres Problem bei der Betrachtung des gesamten Spiels ist, dass dann, wie bereits in 1.1 auf Seite 3 erwähnt, zwei zusätzliche Regeln gelten.

- Falls man im oberen Block der Kniffel-Ziele (1er - 6er) mindesten 63 Punkte erreicht, so erhält man 35 zusätzliche Punkte für das Gesamtspiel.
- Falls man Kniffel bereits erreicht (nicht gestrichen) hat, so hat man das Recht jeden weiteren Kniffel zusätzlich zu den Zielen, deren Definition er erfüllt, auch in ein beliebiges Ziel des unteren Blocks mit fester Punktzahl (full house, kleine oder große Straße) einzutragen, und erhält außerdem, unabhängig vom Erreichen eines Zieles, 100 zusätzliche Punkte für das Gesamtspiel.

Die zweite zusätzliche Regel wirft unsere Annahme über den Haufen, es sei egal ob ein Ziel erreicht oder gestrichen worden ist. Hat man beim einzigen verbleibenden Ziel große Straße den Erstwurf $\{\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{smallmatrix}\}$ erzielt, und ist Kniffel gestrichen, so ist es optimal $\{\begin{smallmatrix} \blacksquare & \blacksquare \end{smallmatrix}\}$ zu behalten, um erwartete $\frac{40 \times 2 \times 6}{216} = \frac{20}{9} \approx 2.2222$ Punkte zu erreichen. Ist jedoch Kniffel bereits erreicht, so behält man besser $\{\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{smallmatrix}\}$ und erzielt $\frac{40+100}{6} = \frac{70}{3} \approx 23.3333$ Punkte. Man muss also nicht 8.192, sondern $3 \times 2^{12} = 12.288$ Spiel-Stellungen betrachten, und auch bei der Bestimmung der Folge-Stellungen vorsichtiger sein.

Die erste Zusatzregel ist jedoch noch schlimmer. Im Prinzip sind alle Punktzahlen $x \in \{0, 1, 2, \dots, 62, 63\}$ im oberen Block als unterschiedlich zu betrachten, ab 63 sind natürlich alle Punktzahlen gleichwertig. Die optimalen Strategien für eine Stellung können dadurch und je nach Punktzahl im oberen Block, komplett verschieden bis entgegengesetzt sein. Beim Erstwurf $\{\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{smallmatrix}\}$ und den Zielen 2er und Kniffel, bedeutet es den Teilwurf $\{\begin{smallmatrix} \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{smallmatrix}\}$ zu behalten, dass man 8.3333 Punkte erhält, $\{\begin{smallmatrix} \blacksquare \end{smallmatrix}\}$ bringt (an sich) nur 3.3333.

Sind also im oberen Block weniger als 53 oder mindestens 63 Punkte eingetragen, so ist erstere Wahl tatsächlich besser. Sind es jedoch 60, so sieht man, ohne alles durchzurechnen, dass man eine Wahrscheinlichkeit von über 50% hat, mit vier Würfeln mindestens eine (zusätzliche) 2 zu erhalten, also ergibt den zweite Teilwurf über $35 \times 0.5 = 17.5000$ Punkte. Aber es könnte sein, dass die exakte Betrachtung der Folge-Stellungen dies erneut umkippen lässt!

Das Ergebnis ist, dass man nicht einen Vektor, sondern eine (12.288×64) -Matrix mit Werten von Folge-Stellungen befüllen muss, genauer genommen sogar zwei solche, eine für Wahrscheinlichkeiten und eine für Erwartungswerte. Wir sprechen von (12.288) Haupt-Stellungen oder groben Stellungen, welche jeweils (64) Unter-Stellungen haben bzw. sich in feine Stellungen unterteilen lassen. Auch gibt es nicht eine Endstellung, sondern $2 \times 64 = 128$ verschiedene, und die Anzahl Stellungen nach x Zügen lässt sich nach einem Pascalschen Dreieck ermitteln (siehe 15). Trotzdem können wir jede feine Spiel-Stellung weiterhin eindeutig durch *HABE* charakterisieren, indem wir -1 für ein gestrichenes Ziel und sonst die Anzahl erzielter Punkte notieren - der Vektor enthält dann sogar einige überflüssige Informationen, wie z.B. die unter Dreierpasch erzielten Punkte.

Alleine durch die Tatsache, dass man Folgestellungen und deren Wert betrachtet, kann sich die Entscheidung ändern, wie man einen konkreten Nachwurf bewertet. Man muss als Folge neben den Elementaren Wahrscheinlichkeiten $W(HABE)$ und Erwartungswerte $E(HABE)$ auch die entsprechenden Größen im Rahmen des Spiels definieren

$$W^{SP}(HABE; PKT) \quad \text{und} \quad E^{SP}(HABE; PKT).$$

Dabei steht SP natürlich für „Spiel“ und PKT sind die im oberen Block bereits eingetragenen Punkte. Da sich PKT in Wirklichkeit aus HABE ergibt, werden wir es nur angeben, falls seine Wichtigkeit zu betonen ist. Dies gilt für alle folgenden Definitionen.

Wir erweitern auf die nahe liegende Art und Weise auch die Definitionen (10) und (11) der optimalen Teilwürfe und der maximalen Werte zu einem Erstwurf, (12) und (13) der optimalen Stellungen-Strategien sowie (14) und (15) der (maximalen) Stellungswerte und verwenden die Bezeichnungen

$$\begin{aligned} OTW^{W,SP}(HABE; PKT; EW) & \quad \text{und} \quad OTW^{E,SP}(HABE; PKT; EW); \\ W^{SP}(HABE; PKT; EW) & \quad \text{und} \quad E^{SP}(HABE; PKT; EW); \\ OS^{W,SP}(HABE; PKT) & \quad \text{und} \quad OS^{E,SP}(HABE; PKT); \\ W_{OS}^{SP}(HABE; PKT) & \quad \text{und} \quad E_{OS}^{SP}(HABE; PKT). \end{aligned}$$

„(Maximaler) Spielstellungswert“ ist der Wert der Stellung an sich, unter Verwendung der optimalen Spielstellungs-Strategie, wobei letztere natürlich von den Werten den Folge-Stellungen abhängt.

Für jede Stellung betrachten wir außerdem den Restwert des Spieles. Dieser entsteht durch Gewichtung der Werte aller (in diesem Zug) erreichbarer Folge-Stellungen. Als Gewichte dienen dabei natürlich die Wahrscheinlichkeiten unter der optimalen Strategie in der jeweiligen Folge-Stellung zu landen. Wir verwenden dafür die Bezeichnungen

$$RW(HABE; PKT) \quad \text{und} \quad RE(HABE; PKT).$$

Indizes wie SP werden nicht benötigt, da die Restwerte nur im Rahmen des Spieles und bei Verwendung optimaler Strategien definiert sind. Der Gesamtstellungswert umfasst sowohl den (maximalen) Spielstellungswert W_{OS}^{SP} bzw. E_{OS}^{SP} wie auch den Restwert und ist gerade der Wert, welcher in die erwähnte Matrix einzutragen ist. Der Wert der Endstellung ist 0, der Restwert des Spieles ist für jede Stellung mit einem einzigen offenen Ziel ebenfalls 0. Somit ist der Gesamtstellungswert dieser „letzter-Zug-Stellungen“ gerade der Spielstellungswert.

4.3 Maschinelle Ermittlung der optimalen (Spiel-) Strategien für alle Stellungen und im Rahmen des Spieles

Genauso wie eine allzu elementare Vorgehensweise bei der Untersuchung aller Erstwürfe, eines Teilwurfs oder einer einzelnen Stellung zu Laufzeit- und/oder Speicherplatz-Problemen führt, kommt man auch beim Spiel insgesamt nicht ohne Weiteres mit dem Kopf durch die Wand. Im Jahr 2006 konnten einige alter Rechner überhaupt keine Stellung bewerten, da zwischendurch an die 130MB Speicherplatz benötigt werden. Auf anderen ergaben sich Laufzeiten von 80s bzw. 20s. Letztere würden damit ca.

$$20 \times 12.288 \times 64 = 15.728.640s \approx 4.370h \approx 182d$$

für die Bewertung aller Stellungen benötigen.

Zwei Verbesserungen wurden eingebaut. Die erste ist mathematischer Natur, etwas komplexer und basiert darauf, dass

1. Nicht jede Unter-Stellung erreichbar ist. Z.B. kann es weder sein, dass man die Ziele 1er, 2er und 5er erreicht hat, und gleichzeitig 0 Punkte im oberen Block hat, noch kann man bereits 60 Punkte in diesem erlangt haben, obwohl man gerade mit dem Spiel beginnt.
2. Nicht alle Unter-Stellungen einer Haupt-Stellung tatsächlich unterscheidbar sind. Dies liegt daran, dass alle sechs Ziele des oberen Blocks diskrete Punktbereiche haben, z.B. können 5er nur 5, 10, 15, 20 oder 25 Punkte ergeben.

Die genaue Implementation ist etwas knifflig (!) und erfolgt in KN_SPIEL_BEWERTEN. Das Prinzip sollte jedoch klar sein. Man markiert einige feine Stellungen als unerreichbar, andere als äquivalent, rechnet jeweils nur einen Repräsentanten letzterer durch, und trägt das Ergebnis in der gesamten Klasse ein. Dadurch muss man „nur“ 310.654 Stellungen bewerten. Zur Bearbeitung wären immer noch ca.

$$20 \times 310.654 = 6.213.080s \approx 1.730h \approx 72d$$

nötig. Die Aufteilung von Stellungen nach x Zügen ist übrigens durch die Aufstellung 15 gegeben. Man beachte, dass die Endstellungen nicht wirklich zu bewerten sind.

Tabelle 15: Anzahl Spiel-Stellungen nach Anzahl erreichter oder gestrichener Ziele, inklusive Zusatzregeln und unter Berücksichtigung äquivalenter feiner Stellungen

Ebene (Züge)	ohne Zusatzregeln	inklusive Kniffel- Regel	inklusive beider Zusatzregeln	erreichbare feine Stellungen	zu bewertende feine Stellungen
Summe	8.192	12.288	786.432	536.448	310.656
0	1	1	64	1	1
1	13	14	896	44	44
2	78	90	5.760	711	704
3	286	352	22.528	5.178	4.900
4	715	935	59.840	21.377	18.875
5	1.287	1.782	114.048	56.283	45.296
6	1.716	2.508	160.512	100.619	71.796
7	1.716	2.640	168.960	126.219	77.034
8	1.287	2.079	133.056	112.596	55.927
9	715	1.210	77.440	71.237	26.770
10	286	506	32.384	31.322	7.940
11	78	144	9.216	9.135	1.282
12	13	25	1.600	1.598	85
13	1	2	128	128	2

Die zweite Verbesserung ist trivial – aber effektiv. Nachdem bereits schon erste (Teil-) Spiel-Bewertungen vorgenommen worden waren, wurde festgestellt, dass sie fehlerhaft waren. Obwohl es nicht die Intention war die KN_*-Funktionen zu optimieren, zeigte sich bei der Suche nach dem Fehler, dass ca. 90%-95% der Laufzeit für den Aufbau aller Nachwürfe aller gewichteten Erstwürfe in KN_STELLUNG_BEWERTEN benötigt wurde. Deshalb speichert nun diese Funktion bei der ersten zu bewertenden Spiel-Stellung alle Nachwürfe in einer Reihe von globalen Hilfsvariablen, welche anschließend verwendet werden können.

Die Bewertung selbst wird durch KN_SPIEL_BEWERTEN vorgenommen. Die Funktion ist schlecht lesbar, unter Anderem wegen der APL2/APL+Win-Syntax, aber auch weil sie andere Elemente, wie z.B. GUI-Steuerung, welche hier nicht interessieren, beinhaltet. Die binären Variablen PROBE, EBENE und NETWORK haben alle den Wert 0 und wir ignorieren sie, genauso wie alle WindowsInterface-Aufrufe. Der Parameter PNR steht auf 1 und ist ebenfalls zu ignorieren. In den globalen Variablen kn_ws bzw. kn_ew werden die Gesamtstellungs-Werte gesammelt. Diese globale Variablen werden in einem Abschnitt der Funktion erzeugt, welcher einmalig, zur Initialisierung, durchlaufen wird und unerreichbare Stellungen markiert:

```
[18] ⌊((2≠⌊NC 'kn_',(ε('ew' 'ws')[1+WS]))^((~(13ρ0)≡HABE)∨(~PROBE)∨(0≠EBENE)))/
'V←(WS 1 0)KN_SPIEL_BEWERTEN(13ρ0)' ⌈ TOTAL-INITIALISIEREN?
[19] →(2=⌊NC 'kn_',(ε('ew' 'ws')[1+WS]))/LO2 ⌈ ERGEBNIS-VARIABLEN VORHANDEN?
[20] ⌊'kn_',(ε('ew' 'ws')[1+WS]),'_stat←14 10ρ0' ⌈ GLOBALE BEWERTUNGS-STATISTIK
: ANZAHL ERREICHTER ZIELE/(GROBE/FEINE/ERREICHBARE/ZU BEWERTENDE/EINZELN Z
U BEWERTENDE) STELLUNGEN/GLOBAL BEWERTETE STELLUNGEN/BEWERTUNGS-ZEIT/LOKAL
BEWERTETE STELLUNGEN/BEWERTUNGS-ZEIT
[21] ⌊'kn_',(ε('ew' 'ws')[1+WS]),'←((3×212),(26))ρ-1' ⌈ GLOBALE WERTE-VARIABL
E: PLATZ FUER ALLE VERSCHIEDENEN SPIELSTELLUNGEN
[22] V←(ι6)∘.×(ι-1ι6) ⌈ MOEGLICHE RESULTATE JEDES DER ERSTEN ZIELE
[23] I←0
[24] L01:
[25] VV←(ϕ(6ρ2)τI)×[1]V ⌈ MOEGLICHE RESULTATE JEDES DER ERSTEN SECHS ZIELE BEI
GEGEBENER STELLUNG
```



```
[26]  ⍺'kn_',(⊃('ew' 'ws')[1+WS]),' [1+I+(26)×(¬1+ι3×26);(¬(¬1+ι64)∈((,',(¬3↓⊃(
    ⊃'VV['),''(⊃''ι6),''(⊃';]∘.+'))),')',((63<+/VV[;6])/',63'),'))/ι64]←¬2' A UNER
    REICHBARE STELLUNGEN (DIE GRENZE 63 KANN AUCH UEBERSCHRITTEN WERDEN)
[27]  I←I+1
[28]  →(I<26)/L01
```

Die Hauptschleife bewertet alle Spielstellungen nach absteigender Ebene, also von den Endstellungen beginnend und mit der Anfangsstellung endend:

```
[42]  V←((11ρ0),¬1 0)+[2]⊞ϕ[1](2 3,(11ρ2))τ(¬1+ι3×212) A GESAMTLISTE: WELCHE ZI
    ELE SIND SCHON GESTRICHEN ODER ERREICHT?
...
[49]  I←13
[50]  L04:
...
[103] L08:
[104] I←I-1
[105] →(I≥EBENE)/L04
```

Innerhalb jeder Ebene werden die groben Stellungen, welche ja Zeilen der Gesamtergebnisse entsprechen, (irgendwie) sortiert und nacheinander abgearbeitet:

```
[51]  NR←(II←I=+/0≠V)≠V A TEILLISTE: I ZIELE GESTRICHEN ODER ERREICHT
[52]  NR←(1+(211)+(2 3,(11ρ2))⊞ϕ[2]NR),[1.1]⊃[2]NR A DURCH ZEILENANGABE ERGAEN
    ZEN
[53]  ⍺'→(0=1↑ρ(NR←(V/¬1=kn_',(⊃('ew' 'ws')[1+WS]),' [NR[;1];])≠NR))/L08' A NOCH
    ZU BEARBEITENDE STELLUNGEN IN DER EBENE I?
[54]  ⍺'NR←',(0=2|PNR[1])/ϕ[1]',NR[⊞(⊞.5×PNR[1])|NR[;1];]' A IN ABHAENGIGKEI
    T VON DER PROZESS-NUMMER SORTIEREN
[55]  ...
[56]  L05:
...
[101] →(0<1↑ρ(NR←1↓[1]NR))/L05
```

Bei den feinen Stellungen wird beschlossen, welche zu derselben Äquivalenzklasse gehören, es ist jeweils nur ein Repräsentant zu bewerten:

```
[59]  ⍺'WERTE←kn_',(⊃('ew' 'ws')[1+WS]),' [NR[1;1];]' A WERTE DER (MOEGLICHERWEIS
    E) ZU UNTERSUCHENDEN STELLUNGEN
[60]  II←(1+2⊞ϕ(⊃NR[1;2])[ι6])⊃KANN A RELEVANTE STELLUNGEN
[61]  VV←(¬(⊃NR[1;2])[ι6])×[1](ι6)∘.×(¬1+ι6) A MOEGLICHE RESULTATE JEDES DER NOC
    H VERBLEIBENDEN UNTER DEN ERSTEN SECHS ZIELE
[62]  ⍺'VV←',(¬3↓⊃(⊃'VV['),''(⊃''ι6),''(⊃';]∘.+')) A MOEGLICHE RESULTATE DER NOCH
    VERBLEIBENDEN UNTER DEN ERSTEN SECHS ZIELE INSGESAMT
[63]  ⍺'VV←(63-(¬1+ι63)∈VV)/(¬1+ι63)',(V/63≤VV)/',0') A BEZUEGLICH DER GRENZE
    63 UNTERSCHIEDLICHE FEINE STELLUNGEN
[64]  II←((ρVV),ιρVV)[1++/[1]VV∘.>II],[1.1]II A RELEVANTE STELLUNGEN GRUPPIEREN
    UND
[65]  II←(VV[;1]∈II[;1])≠(VV←((ρVV),ιρVV)[1++/[1]VV∘.>(¬1+ι64)],[1.1](¬1+ι64)) A
    ERGAENZEN
[66]  →(0=1↑ρ(II←(¬1=WERTE[1+II[;2]])≠II))/L07 A ZU BEWERTENDE (RELEVANTE) STELL
    UNGEN UND
...
[71]  L06:
...
[97]  L07:
[98]  ⍺(0<1↑ρII)/'II←(II[1;1]≠II[;1])≠II'
[99]  →(0<1↑ρII)/L06
```

Im Hauptteil wird im Wesentlichen KN_STELLUNG_BEWERTEN aufgerufen und das Resultat global notiert:

```
[72] VV←'1 1 1 1 1 1 7 6 25 30 40 50 5' ⍲ STANDARD-WERTE DER VERSCHIEDENEN
    ZIELE
[73] ⍺((√/(>NR[1;2]))[16])^(0≠II[1;2]))/VV[⌊(>NR[1;2])[16]/16]←II[1;2]'
[74] WERT←(5ρ0)(,0)('')(0 14ρ0)
[75] ...
[76] ⍺((13≠I)^(~PROBE))/'→(0≡(WERT←(WS 1)KN_STELLUNG_BEWERTEN('1[(>NR[1;2])×VV)
    ))/ENDE' ⍲ STELLUNG (INKLUSIVE RESTSPIEL) BEWERTEN
[77] ⍺'WERTE[1+(II[1;1]=II[;1])/II[;2]]←+/,((2>WERT)×[1](4>WERT))÷(+/2>WERT)' ⍲
    (GEWICHTETE UND NORMIERTE) BEWERTUNG LOKAL NOTIEREN
...
[82] ⍺(~PROBE)/'kn_',(ε('ew' 'ws')[1+WS]),' [NR[1;1];]←WERTE' ⍲ VORGENOMENE BEWE
    RTUNGEN GLOBAL NOTIEREN?
```

Dadurch, dass 1 als Parameter *SPIEL* übergeben wird, ändert sich der Algorithmus dieser Funktion, und damit auch KN_ERSTWURF_BEWERTEN und KN_WURF_BEWERTEN, etwas, es werden an mehreren Stellen die Folgestellungen berücksichtigt und es wird jeweils beschlossen, welches Ziel bei Misserfolg zu streichen ist.

4.4 Beispiele (maximaler) Spielstellungs-Wahrscheinlichkeiten und -Erwartungswerte sowie optimaler Spielstrategien

Damit können nun optimale Spielstrategien ermittelt und Resultate verglichen werden. Wir untersuchen erst noch ein Mal die Einzelziele. Der Vergleich mit den Ergebnissen aus 1 und 10 ist nicht trivial, da hier die Zusatzregeln greifen. Wir untersuchen vier Varianten: mit Kniffel gestrichen (1 und 3) bzw. erzielt (2 4), kombiniert mit einer oberen Summe innerhalb vier (1 und 2) bzw. eines (3 und 4) Zielwürfeln von 63. Damit ist gemeint, dass, für das Ziel 4er z.B., angenommen wird, dass die Summe der Punkte der restlichen fünf Ziele des Blocks zwischen 47 und 50 bzw. zwischen 59 und 62 liegt. Bei Kniffel selbst als Ziel ist dieses Ziel natürlich stets offen.

Die Aufstellung 16 zeigt, wie zu erwarten, dass die Wahrscheinlichkeit ein Ziel mit variabler Punktzahl zu erreichen nicht von den Zusatzregeln abhängt. Für die Ziele mit fester Punktzahl steigert ein vorhandener Kniffel die Erfolgswahrscheinlichkeit.

Tabelle 16: Spielstellungs-Wahrscheinlichkeiten (in Prozent) für alle einzelnen Kniffel-Ziele und für vier Zusatzregeln-Szenarien

HABE und PKT je nach Ziel und Szenario						
Ziel	W(.)	W _{OS} (.)	W _{OS} ^{SP} (.;.)			
			Sz. 1	Sz. 2	Sz. 3	Sz. 4
1er	59.81	83.85	83.85	83.85	83.85	83.85
2er	59.81	83.85	83.85	83.85	83.85	83.85
3er	59.81	83.85	83.85	83.85	83.85	83.85
4er	59.81	83.85	83.85	83.85	83.85	83.85
5er	59.81	83.85	83.85	83.85	83.85	83.85
6er	59.81	83.85	83.85	83.85	83.85	83.85
Dreierpasch	21.30	54.13	54.13	54.13	54.13	54.13
Viererpasch	2.01	13.23	13.23	13.23	13.23	13.23
full house	3.86	19.12	19.12	19.41	19.12	19.41
kleine Straße	15.43	41.18	41.18	41.27	41.18	41.27
große Straße	3.09	12.97	12.97	13.29	12.97	13.29
Kniffel	0.08	1.26	1.26	1.26	1.26	1.26
Chance	100.00	100.00	100.00	100.00	100.00	100.00

Die Aufstellung 17 zeigt die volle Auswirkung der Zusatzregeln auf die Erwartungswerte. Die Punkte im oberen Block hängen stark von der Nähe zur 63-Punkte-Grenze ab, jedoch auch vom Vorhandensein eines Kniffels. Der untere Block wird nur von letzterem beeinflusst.

Tabelle 17: Spielstellungs-Erwartungswerte (in Punkten) für alle einzelnen Kniffel-Ziele und für vier Zusatzregeln-Szenarien

HABE und PKT je nach Ziel und Szenario						
Ziel	$W(.)$	$W_{OS}(.)$	$W_{OS}^{SP}(.; .)$			
			Sz. 1	Sz. 2	Sz. 3	Sz. 4
1er	0.8333	1.5278	2.6803	2.9467	30.8751	31.1414
2er	1.6667	3.0556	4.2081	4.4745	32.4029	32.6692
3er	2.5000	4.5833	5.7359	6.0022	33.9306	34.1970
4er	3.3333	6.1111	7.2637	7.5300	35.4584	35.7248
5er	4.1667	7.6389	8.7915	9.0578	36.9862	37.2525
6er	5.0000	9.1667	10.3192	10.5856	38.5140	38.7803
Dreierpasch	3.7269	10.2087	10.2087	11.1830	10.2087	11.1830
Viererpasch	0.3511	2.4215	2.4215	3.6696	2.4215	3.6696
full house	0.9645	4.7806	4.7806	5.9013	4.7806	5.9013
kleine Straße	4.6296	12.3536	12.3536	12.7641	12.3536	12.7641
große Straße	1.2346	5.1899	5.1899	6.0287	5.1899	6.0287
Kniffel	0.0386	0.6316	0.6316	0.6316	0.6316	0.6316
Chance	17.5000	21.2500	21.2500	21.8073	21.2500	21.8073

Das zweite Beispiel, welches wir von Beginn an verfolgt haben, ist eine Kombination von vier Zielen. Da das Spiel hier einen Restwert hat, ist es sinnvoll auch W mit W^{SP} sowie E mit E^{SP} zu vergleichen. Da Kniffel offen ist, sind nur zwei Szenarien, welche nach Punkten im oberen Block differenzieren, zu betrachten. Interessant ist, dass es sich im Rahmen des Spieles lohnt, ein paar 1er zu Gunsten von full house und große Straße zu opfern: Der Gesamterfolg im Zug und bei optimaler Strategie ist niedriger, dadurch wird jedoch eine bessere Folgestellung für das Restspiel erkaufte.

Tabelle 18: Spielstellungs-Wahrscheinlichkeiten (in Prozent) für eine Kombination von vier Kniffel-Zielen und für zwei Zusatzregeln-Szenarien

HABE und PKT je nach Ziel und Szenario						
Ziel	$W(.)$	$W_{OS}(.)$	Szenario 1		Szenario 2	
			$W^{SP}(.)$	$W_{OS}^{SP}(.; .)$	$W^{SP}(.)$	$W_{OS}^{SP}(.; .)$
Summe	63.99	87.03	63.99	76.85	63.99	76.85
1er	59.81	81.35	56.97	51.44	56.97	51.44
full house	2.57	3.50	3.86	15.50	3.86	15.50
große Straße	1.54	2.10	3.09	9.26	3.09	9.26
Kniffel	0.06	0.09	0.08	0.65	0.08	0.65
Restwert			103.90	105.27	103.90	105.27

Bei den Erwartungswerten greift die erste Zusatzregel wieder voll. Hier lohnt es sich auch nur 1er einzutragen, welche zur Überschreitung der Grenze führen. Dies sieht man an der im Rahmen des Spieles niedrigeren Wahrscheinlichkeit das Ziel ohne Nachwurf zu erreichen.

Tabelle 18: Spielstellungs-Wahrscheinlichkeiten (in Prozent) für eine Kombination von vier Kniffel-Zielen und für zwei Zusatzregeln-Szenarien

HABE und PKT je nach Ziel und Szenario						
			Szenario 1		Szenario 2	
Ziel	$W(\cdot)$	$W_{OS}(\cdot)$	$W^{SP}(\cdot)$	$W_{OS}^{SP}(\cdot; \cdot)$	$W^{SP}(\cdot)$	$W_{OS}^{SP}(\cdot; \cdot)$
Summe	63.99	87.03	63.99	76.85	63.99	76.85
1er	59.81	81.35	56.97	51.44	56.97	51.44
full house	2.57	3.50	3.86	15.50	3.86	15.50
große Straße	1.54	2.10	3.09	9.26	3.09	9.26
Kniffel	0.06	0.09	0.08	0.65	0.08	0.65
Restwert			103.90	105.27	103.90	105.27

Bei den Erwartungswerten greift die erste Zusatzregel wieder voll. Hier lohnt es sich auch nur 1er einzutragen, welche zur Überschreitung der Grenze führen. Dies sieht man an der im Rahmen des Spieles niedrigeren Wahrscheinlichkeit das Ziel ohne Nachwurf zu erreichen.

Tabelle 19: Spielstellungs-Erwartungswerte (in Punkten) für eine Kombination von vier Kniffel-Zielen und für zwei Zusatzregeln-Szenarien

HABE und PKT je nach Ziel und Szenario						
			Szenario 1		Szenario 2	
Ziel	$E(\cdot)$	$E_{OS}(\cdot)$	$E^{SP}(\cdot)$	$E_{OS}^{SP}(\cdot; \cdot)$	$E^{SP}(\cdot)$	$E_{OS}^{SP}(\cdot; \cdot)$
Summe	3.0228	8.7085	2.7488	8.8969	22.9623	26.7655
1er	0.7851	0.7353	0.5112	0.8272	20.7247	18.8280
full house	0.9645	3.5365	0.9645	3.6330	0.9645	3.5722
große Straße	1.2346	4.0123	1.2346	4.0123	1.2346	4.0123
Kniffel	0.0386	0.4244	0.0386	0.4244	0.0386	0.3529
Restwert			20.9965	20.1038	34.4556	35.4182

Der Vollständigkeit halber analysieren wir in 20 auch den Erstwurf $\{\cdot, \cdot, \cdot, \cdot, \cdot\}$ für dieselbe Kombination von Zielen und unter den zwei Szenarien.

Tabelle 20: Optimale Teilwürfe im Rahmen des Spieles für eine Kombination von vier Kniffel-Zielen und für zwei Zusatzregeln-Szenarien






HABE und PKT je nach Ziel und Szenario, EW = $\{\cdot, \cdot, \cdot, \cdot, \cdot\}$						
		Sz.1	Sz. 2		Sz. 1	Sz. 2
Ziel	$W(\cdot; \text{EW})$	$W^{SP}(\cdot; \cdot; \text{EW})$		$E^{SP}(\cdot; \text{EW})$	$E(\cdot; \cdot; \text{EW})$	
OTW	$\{\cdot\}$	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$	$\{\cdot, \cdot, \cdot, \cdot, \cdot\}$
Summe	100.00	100.00	100.00	9.6667	9.6667	33.0000
1er	100.00	66.67	66.67	1.3333	1.3333	24.6667
full house	0.00	33.33	33.33	8.3333	8.3333	8.3333
große Straße	0.00	0.00	0.00	0.0000	0.0000	0.0000
Kniffel	0.00	0.00	0.00	0.0000	0.0000	0.0000
Restwert		87.83	87.83		19.1249	30.0751

Als letztes Ergebnis muss der Wert des Spieles bestimmt werden. Beim ersten Zug gilt stets PKT = 0 und Kniffel ist offen. Man kann also das Ergebnis des Erstwurfs, den Nachwurf zum speziellen Erstwurf $\{\square, \square, \square, \square, \square\}$ und den Wert der Stellung, jeweils als einzelne Stellung sowie im Rahmen des Spieles, direkt vergleichen.

Tabelle 21: Analyse der Anfangs-Stellung von Kniffel bezüglich Wahrscheinlichkeiten

HABE = (0, 0, 0, 0, 0, 0 ; 0, 0, 0, 0, 0, 0), EW = $\{\square, \square, \square, \square, \square\}$, alle Wahrscheinlichkeiten in %						
	$OTW^W(\text{HABE}; \{EW\}) = \emptyset$			$OTW^{W,SP}(\text{HABE}; \{EW\}) = \{\square, \square, \square, \square, \square\}$		
Ziel	$W(\text{HABE})$	$W(\text{HABE}; EW)$	$W_{OS}(\text{HABE})$	$W^{SP}(\text{HABE})$	$W^{SP}(\text{HABE}; EW)$	$W_{OS}^{SP}(\text{HABE})$
1er	50.15	50.15	50.15	42.44	66.67	24.63
2er	20.83	20.83	20.83	0.00	0.00	0.00
3er	6.56	6.56	6.56	3.47	0.00	2.89
4er	1.16	1.16	1.16	1.16	0.00	0.96
5er	0.00	0.00	0.00	0.00	0.00	0.00
6er	0.00	0.00	0.00	16.20	0.00	13.31
Dreierpasch	19.29	19.29	19.29	15.43	0.00	13.72
Viererpasch	2.01	2.01	2.01	1.93	0.00	7.18
full house	0.00	0.00	0.00	3.86	33.33	15.43
kleine Straße	0.00	0.00	0.00	12.35	0.00	11.70
große Straße	0.00	0.00	0.00	3.09	0.00	9.26
Kniffel	0.00	0.00	0.00	0.08	0.00	0.91
Chance	0.00	0.00	0.00	0.00	0.00	0.00
Summe	100.00	100.00	100.00	100.00	100.00	100.00
Restwert	0.00	0.00	0.00	1035.78	1039.39	1043.55
Spielwert	0.00	0.00	0.00	1135.78	1139.39	1143.55

Tabelle 22: Analyse der Anfangs-Stellung von Kniffel bezüglich Erwartungswerten

HABE = (0, 0, 0, 0, 0, 0 ; 0, 0, 0, 0, 0, 0), EW = {  ,  ,  ,  ,  }, alle Erwartungswerte in Punkten						
	$OTW^E(\text{HABE}; \{EW\}) = \{\langle \cdot \rangle\}$			$OTW_{E,SP}(\text{HABE}; \{EW\}) = \{\langle \cdot \rangle, \langle \cdot \rangle, \langle \cdot \rangle, \langle \cdot \rangle\}$		
Ziel	$E(\text{HABE})$	$E(\text{HABE}; EW)$	$E_{OS}(\text{HABE})$	$E^{SP}(\text{HABE})$	$E^{SP}(\text{HABE}; EW)$	$E_{OS}^{SP}(\text{HABE})$
1er	0.0000	0.0000	0.0000	0.3781	0.0000	0.1575
2er	0.0000	0.0000	0.0000	0.7253	2.6667	0.4391
3er	0.0000	0.0000	0.0000	0.7870	0.0000	0.6145
4er	0.0000	0.0000	0.0000	0.8333	0.0000	1.0110
5er	0.0000	0.0000	0.0000	0.9002	0.0000	1.3558
6er	0.0000	0.0000	0.0000	0.8642	0.0000	1.3062
Dreierpasch	2.8048	3.2917	3.1780	0.6096	0.0000	1.3834
Viererpasch	0.3376	0.4267	0.6645	0.0965	0.0000	0.4359
full house	0.8681	0.7716	1.0858	0.9324	8.3333	2.5024
kleine Straße	3.7037	3.3333	7.6888	3.7037	0.0000	4.7261
große Straße	1.2346	1.4815	3.8670	1.2346	0.0000	2.9150
Kniffel	0.0386	0.0386	0.1857	0.0386	0.0000	0.5298
Chance	11.0725	11.8056	8.5971	1.6860	0.0000	1.5817
Summe	20.0598	21.1489	25.2669	12.7894	11.0000	18.9584
Restwert	0.0000	0.0000	0.0000	172.9645	176.0446	171.0149
Spielwert	0.0000	0.0000	0.0000	185.7539	187.0446	189.9733

Was nicht ermittelt wurde, ist die Verteilung des Gesamtwerts des Spieles auf einzelnen Ziele. Dazu müsste der Wert jeder Stellung nicht als Summe, sondern als Vektor der Länge 13 gespeichert werden. Vielleicht wird dies in einer zukünftigen Version gemacht.

A Ergänzendes Material

A.1 Oberfläche zur Steuerung der Hauptfunktionen und zur Anzeige der Ergebnisse, parallele Bewertung von Kniffel durch mehrere Prozesse

Die Hauptfunktionen zur Bewertung von Kniffel wurden im Artikel grob beschrieben. Man kann sie natürlich direkt in APL+Win starten. Sie können jedoch auch über eine sehr einfache Oberfläche (das Ergebnis eines interaktiven Kennenlernens des Konzeptes „GUI“) gesteuert werden. Mittels der globalen Variablen `kn_gui` kann man den Bewertungsfunktionen mitteilen, ob das GUI offen ist oder nicht.

In dem snapshot 1 kann man (mit etwas Mühe!) die Ergebnisse zu Erwartungswerten aus den Tabellen 19 und 20 erkennen. Genauer findet man den Gesamtwert der Einzel-/Spiel-Stellung bei optimaler Strategie und bei Verwendung des zweiten Szenarios (man beachte die Summe der Punkte in der oberen Hälfte) sowie der Restwert, und die Bewertung des Erstwurfs, sowohl für die Stellung an sich wie auch im Rahmen des Spieles. Außerdem sieht man die Verteilung der Erwartungswerte bei optimaler Strategie im Rahmen des Spieles auf die einzelnen Ziele.

Die Oberfläche wird durch `KN_GUI` aufgebaut. Folgende Funktionen stellen (mehr oder minder) Event Handlers dar: `KN_CHANGE_POSITION`, `KN_DO`, `KN_PLACE`, `KN_RESET_POSITION`, `KN_RESIZE`, `KN_SHOW_MORE` und `KN_SHOW_RESULT`. `KN_EXCEL` enthielt erste Überlegungen zum Export der Ergebnisse nach Excel, wurde aber nicht weiterentwickelt. Der code dieser Funktionen wird nicht angegeben, kann aber bei Bedarf zur Verfügung gestellt werden.

Ein weiterer Punkt ist in den Hauptfunktionen sichtbar: Aufrufe von weiteren Unterfunktionen, welche sich mit „Kniffel-Netzen“ befassen. Da die ursprünglichen Schätzungen für die Laufzeit der gesamten Spielanalyse sehr hoch waren, wurde überlegt wie man das Problem parallelisieren kann. Dies ist auf Ebene von Stellungen sehr einfach, jede solche kann für sich aus bearbeitet werden, sofern alle Folgestellungen bereits bewertet worden sind.

Daraus wurde eine Übung zur gemeinsamen Nutzung von component files gemacht. Sofern man die Bewertung über die Oberfläche startet, werden nicht nur die Hauptresultate in einem solchen cf abgelegt, sondern es werden auch Angaben zum Problem, an dem gearbeitet wird, in Hilfsdateien gespeichert. Zu Beginn der Bewertung schaut das Programm nach, ob bereits weitere Prozessoren daran arbeiten. Falls ja, holt sich das Programm die nächste freie Nummer und meldet sich am Netz an.

Das eigentliche Problem ist immer die Bewertung des gesamten Spiels, es kann aber auch nur ein Teil davon bewertet werden. Stellungen werden nach absteigender Ebene (Anzahl erfolgter Züge) bearbeitet. Innerhalb einer Ebene wählt jeder Prozess die Sortierung nach seiner Nummer im Netz, und zwar so, dass möglichst wenig Mehrfachbearbeitung von Stellungen stattfindet.

Folgende Funktionen befassen sich mit der Parallelisierung: `KN_SAVE_RESULTS`, `KN_SHARE_FILE`, `KN_SHARE_RESULTS` und `KN_ATTENTION`. Auch deren code wird nicht angegeben, kann jedoch zur Verfügung gestellt werden.

A.2 Der code der zur Analyse des Spieles verwendeten Funktionen

Die eigentliche Analyse von Kniffel wird durch fünf Funktionen erledigt. Es handelt sich um die Bewertungsfunktionen `KN_SPIEL_BEWERTEN`, `KN_STELLUNG_BEWERTEN`, `KN_ERSTWURF_BEWERTEN` und `KN_WURF_BEWERTEN`, sowie `KN_WURF_GEWICHTEN`. Die vier ersten Funktionen lassen sich, so wie angegeben, streng hierarchisch anordnen. Die letzte wird als „Sackgasse“ aus allen Ebenen außer der untersten aufgerufen.

Zumindest teilweise wurde deren code schon vorgestellt, genauer die unbedingt zur Funktionalität notwendigen Zeilen. Es folgt der vollständige code. Er wurde nach 130 Zeichen umgebrochen, Folgezeilen beginnen mit Nabla. Der workspace an sich kann zur Verfügung gestellt werden, auch als runtime version.

Programm zur Analyse von "Kniffel" - Version 1.5

Spiel-Stellung			Bewertung als Einzelzug		Bewertung im Rahmen des Spiels	
			Stellung bewerten		Stellung bewerten	
1er	<input type="checkbox"/>	18.83 Punkte!	<input type="button" value="Los!"/>	<input type="checkbox"/> Ws	<input type="button" value="Los!"/>	<input type="checkbox"/> Ws
2er	<input type="checkbox"/> 8	0.00 Punkte!	Stellungs-Wert: 8.71 Punkte!		Stellungs-Wert: 26.77 Punkte!	
3er	<input type="checkbox"/> 9	0.00 Punkte!	Strategie (Erstwurf/zu behalten/Wert)		Strategie (Erstwurf/zu behalten/Wert)	
4er	<input type="checkbox"/> 12	0.00 Punkte!	1 1 1 1 1/1 1 1 1 1/50		1 1 1 1 1/1 1 1 1 1/50	
5er	<input type="checkbox"/> 15	0.00 Punkte!	2 2 2 2 2/2 2 2 2 2/50		2 2 2 2 2/2 2 2 2 2/50	
6er	<input type="checkbox"/> 18	0.00 Punkte!	3 3 3 3 3/3 3 3 3 3/50		3 3 3 3 3/3 3 3 3 3/50	
Dreierpasch	<input type="checkbox"/> 16	0.00 Punkte!	Restspiel-Wert: 35.42 Punkte!		Restspiel-Wert: 35.42 Punkte!	
Viererpasch	<input type="checkbox"/> 22	0.00 Punkte!	Erstwurf bewerten		Erstwurf bewerten	
full house	<input type="checkbox"/>	3.57 Punkte!	<input type="button" value="Los!"/> <input type="checkbox"/> Ws.		<input type="button" value="Los!"/> <input type="checkbox"/> Ws.	
kleine Strasse	<input type="checkbox"/> 30	0.00 Punkte!	1 1 2 2 5		1 1 2 2 5	
grosse Strasse	<input type="checkbox"/>	4.01 Punkte!	1 1 2 2 behalten!		1 1 2 2 behalten!	
Kniffel	<input type="checkbox"/>	0.35 Punkte!	Zweitwurf-Wert: 9.66 Punkte!		Zweitwurf-Wert: 33.00 Punkte!	
Chance	<input type="checkbox"/> 20	0.00 Punkte!	Restspiel-Wert: 30.08 Punkte!		Restspiel-Wert: 30.08 Punkte!	
			Wurf bewerten		Wurf bewerten	
			<input type="button" value="Los!"/> <input type="checkbox"/> Ws.		<input type="button" value="Los!"/> <input type="checkbox"/> Ws.	
			1 1 2 2 5		1 1 2 2 5	
			1er!		1er!	
			Restspiel-Wert: 20.29 Punkte!		Restspiel-Wert: 20.29 Punkte!	

Zuruecksetzen Beenden

Abbildung 1: Die Hauptoberfläche zur Analyse von Kniffel und der interaktiven Darstellung der Ergebnisse

CharacterRepresentation von KN_SPIEL_BEWERTEN:

```

ERG←STEUER KN_SPIEL_BEWERTEN HABE;WS;PROBE;EBENE;NETWORK;ALX;NNR;PNR;TS;NR;KANN;TAUSCH;WERTE;AI;V;VV;VW;I;II;III
A
A BEWERTET ALLE, VON DER ANGEBOGENEN AUS ERREICHBAREN, KNIFFEL-SPIELSTELLUNGEN
A EBENE -1 ZUR UEBERGABE DER (LETZTEN) BEWERTUNG WAHLEN
A
A INITIALISIEREN
A
STEUER←4↑(STEUER, '0')[1+2#NC 'STEUER'] A STANDARD: ALLE ERWARTUNGSWERTE OHNE HILFE BERECHNEN
(WS PROBE EBENE NETWORK)←STEUER
EBENE←EBENE[(+/O#HABE) A MINIMALE NOETIGE BEWERTUNGS-EBENE
kn_gui←(ε('kn_gui', '4=NC 'AV','))[1+2#NC 'kn_gui'] A STANDARD-UMGEBUNG IN ABHAENGIGKEIT VOM AV
(ε(kn_gui)/[wself←'fm_KN_MAIN', ε ALX←'KN_ATTENTION -1', ((v/'' ''#ALX)/'' ε '' ,ALX) ε WS KN_SAVE_RESULTS -1' A GEGEBENFALLS
  ▽ GESPEICHERTE ERGEBNISSE EINLESEN (DUMMY FUER FEHLER BEI FILE-OPERATIONEN)
PNR←NNR+1 0 ε TS←0 ε NR←0 2p0 ε TAUSCH←5↑0 1 A BEI ANMELDEN AN EIN NETZWERK UEBERPRUEFEN WELCHE ZEILEN IN BEARBEITUNG SIND
(ε(kn_gui^(~PROBE)^(1=NETWORK)))/(NNR PNR TS)←(HABE WS PNR NNR) KN_SHARE_FILE 1 ε NR←(WS NR TAUSCH NNR PNR TS) KN_SHARE_RESULTS -1'
  ▽ A GEGEBENFALLS AN EIN NETZWERK ANMELDEN
(ε(kn_gui)/[ALX←(2++/1>+\\', ε ''=ALX)↑ALX' A DUMMY-NOTSPEICHERUNG ENTFERNEN
A
A GLOBALE STATISTIK- UND WERTE-VARIABLEN BEREITSTELLEN
A
(ε((2#NC 'kn_', (ε('ew', 'ws')[1+WS]))^(~(13p0)≡HABE)^(~PROBE)^(O#EBENE)))/V←(WS 1 0)KN_SPIEL_BEWERTEN(13p0)' A INITIALISIEREN?
→(2=NC 'kn_', (ε('ew', 'ws')[1+WS]))/L02 A ERGEBNIS-VARIABLEN VORHANDEN?
  ε'kn_', (ε('ew', 'ws')[1+WS]), '_stat←14 10p0' A GLOBALE BEWERTUNGSSTATISTIK: ANZAHL ERREICHTER ZIELE/(GROBE/FEINE/ERREICHBARE/ZU BEW
  ▽ ERLENDE/EINZELN ZU BEWERTENDE) STELLUNGEN/GLOBAL BEWERTETE STELLUNGEN/BEWERTUNGS-ZEIT/LOKAL BEWERTETE STELLUNGEN/BEWERTUNGSZEIT
  ε'kn_', (ε('ew', 'ws')[1+WS]), '←((3×212), (26))p -1' A GLOBALE WERTE-VARIABLE: PLATZ FUER ALLE VERSCHIEDENEN SPIELSTELLUNGEN
V←(16)ε . x (1+16) A MOEGLICHE RESULTATE JEDES DER ERSTEN ZIELE
I←0
L01:
VW←(φ(6p2)↑I)×[1]V A MOEGLICHE RESULTATE JEDES DER ERSTEN SECHS ZIELE BEI GEGEBENER STELLUNG
  ε'kn_', (ε('ew', 'ws')[1+WS]), '[1+I+(26)×(1+13×26)]; (~(-1+164)ε((, '(-3+ε(c'VV[', '' t6), ''(c';]ε.+.)), ''((63<+/VV[;61])/', 63')), '
  ▽ ))/164]←-2' A UNERREICHBARE STELLUNGEN (DIE GRENZE 63 KANN AUCH UEBERSCHRITTEN WERDEN)
I←I+1
→(I<26)/L01
A
A RELEVANTE SPIELSTELLUNGEN BESTIMMEN
A
L02:
  ε(1+ε, (~1=3÷STEUER)×(~PROBE)×(~2 -1 t kn_', (ε('ew', 'ws')[1+WS]), '[1+((2 3, (11p2))↑φ((O#HABE[111]), (~1+(-1 0 50 t HABE[12])), (O#HABE[13

```

```

▽ ]))];1++/0[HABE[t6]]))>'' ''UNERREICHBARE SPIELSTELLUNG, NICHTS ZU BEWERTEN, ABRUCH'' ♦ ERG←0 ♦ →0'' ''SCHON BEWERTETE SP
IELSTELLUNG, NICHTS ZU BEWERTEN, ABRUCH'' ♦ ERG←0 ♦ →0'
KANN←0p0 A VON DER GEGEBENEN STELLUNG AUS ERREICHBARE STELLUNGEN
V←(0=HABE[t6])×[1](t6)◦.×(¬(1+t6) A MOEGLICHE RESULTATE JEDES DER NOCH VERBLEIBENDEN UNTER DEN ERSTEN SECHS ZIELE
I←0
L03:
VV←(φ(6p2)τI)×[1]V A MOEGLICHE RESULTATE JEDES DER ERSTEN SECHS ZIELE BEI GEGEBENER STELLUNG
# KANN←KANN,(c(¬(1+t64)ε((+/0[HABE[t6]]+<.,'(-3+ε(c'VV[''],'(¬t6),''(c';]◦.+')),'((63<(-+/0[HABE[t6]])+(+/VV[6]))/','63'))/('¬
V 1+t64)))' A ERREICHBARE STELLUNGEN (DIE GRENZE 63 KANN AUCH UEBERSCHRITTEN WERDEN)
I←I+1
→(I<26)/L03
V←((11p0),¬1 0)+[2]qφ[1](2 3,(11p2))τ(¬1+t3×212) A GESAMTLISTE: WELCHE ZIELE SIND SCHON GESTRICHEN ODER ERREICHT?
V←(^(0=HABE)×[2]((0#HABE[t11]),(¬2+¬1 0 50tHABE[t12]), (0#HABE[t13]))=[2]V))×[1]V A VON DER ANGEgebenEN STELLUNG AUS ERREICHBARE
V BEHALTEN
A
A ERREICHBARE UND RELEVANTE SPIELSTELLUNGEN VOM SPIELLENDE HER BEWERTEN
A
#(kn_gui^(~PROBE))/[ALX←''(WS TAUSCH NNR PNR) KN_ATTENTION NETWORK'',((v/'' ''#□ALX)/'' ♦ ''',□ALX)' A BEI UNTERBRECHUNG IN WINDOW
V S-UMGEBUNG ERGEBNISSE NOTSPEICHERN
ERG←14 8p0 A LOCALE BEWERTUNGS-STATISTIK: ANZAHL ERREICHTER ZIELE/(GROBE/FEINE/ERREICHBARE/ZU BEWERTENDE/EINZELN ZU BEWERTENDE) ST
ELLUNGEN/LOKAL BEWERTETE STELLUNGEN/BEWERTUNGS-ZEIT
I←13
L04:
NR←(II←I++/0v)÷V A TEILLISTE: I ZIELE GESTRICHEN ODER ERREICHT
NR←(1+(211)+(2 3,(11p2))1qφ[2]NR),[1.1]c[2]NR A DURCH ZEILENANGABE ERGAENZEN
#'→(0=1tp(NR←(v/¬1=kn_',(ε('ew' 'ws')[1+WS]),'[NR[1];]÷NR))/L08' A NOCH ZU BEARBEITENDE STELLUNGEN IN DER EBENE I?
#'NR←',((0=2|PNR[1])/'φ[1]'),'NR[(f.5×PNR[1])|NR[1];]';] A IN ABHAENGIGKEIT VON DER PROZESS-NUMMER SORTIEREN
#'ERG[1+I;]←I,(1tpNR),((26)×1tpNR),(+/¬2#εII÷kn_',(ε('ew' 'ws')[1+WS]),')0 0 0,□AI[2+~kn_gui]' A NOCH KEINE STELLUNG BEWERTET/BE
WERTUNGS-BEGINN
L05:
#(kn_gui^(~PROBE)^(1=NETWORK)^(0#TAUSCH[2]))/NR←(WS NR TAUSCH NNR PNR TS) KN_SHARE_RESULTS ¬2 ♦ →(0=1tp(NR←(v/¬1=kn_',(ε('ew' 'ws
V '')[1+WS]),'[NR[1];]÷NR))/L08' A GEGEBENFALLS ERGEBNISSE EINHOLEN - NOCH ZU BEARBEITENDE STELLUNGEN IN DER EBENE I?
TAUSCH←5t I
#WERTE←kn_',(ε('ew' 'ws')[1+WS]),'[NR[1];]';] A WERTE DER (MOEGLICHERWEISE) ZU UNTERSUCHENDEN STELLUNGEN
II←(1+21qφ(¬NR[1;2]) [t6])>KANN A RELEVANTE STELLUNGEN
VV←(¬(¬NR[1;2]) [t6])×[1](t6)◦.×(¬1+t6) A MOEGLICHE RESULTATE JEDES DER NOCH VERBLEIBENDEN UNTER DEN ERSTEN SECHS ZIELE
#VV←',(-3+ε(c'VV[''],'(¬t6),''(c';]◦.+'))
#VV←(63-((¬1+t63)εVV)/(¬1+t63))',((v/63≤VV)/'0') A BEZUEGLICH DER GRENZE 63 UNTERSCHIEDLICHE FEINE STELLUNGEN
II←((pVV),tpVV)[1++/[1]VV◦.>II],[1.1]II A RELEVANTE STELLUNGEN GRUPPIEREN UND

```

47

```

→ (0<1↑ρ(NR←1↓[1]NR))/L05
ERG[1+I;8]←[AI[2+~kn_gui]-ERG[1+I;8]
L08:
I←I-1
→ (I≥EBENE)/L04
A
A ERGEBNIS-VARIABLEN GLOBAL SPEICHERN, BEENDEN
A
ENDE:
A (((13ρ0)≡HABE)∧PROBE∧(0=EBENE))/'kn_',(ε('ew' 'ws')[1+WS]),'_stat[;16]←ERG[;16]' A TOTAL-INITIALISIERUNG?
A ((~kn_gui)∧(~PROBE))/'kn_',(ε('ew' 'ws')[1+WS]),'_stat←kn_',(ε('ew' 'ws')[1+WS]),'_stat[I;16],(kn_',(ε('ew' 'ws')[1+WS])),'_stat[I
∇ ;7 8]+((1≠SERVER)×kn_',(ε('ew' 'ws')[1+WS]),'_stat[I;9 10]),(14 2ρ0)' A STELLUNGEN BEWERTET?
A ((~1=3→STEUER)∧(~PROBE))/'ERG←WERT' A BEWERTUNG UEBERGEHEN?
→ (~kn_gui)/0
A ((~PROBE)∧(1=NETWORK))/'(NNR PNR TS)←(HABE WS NNR PNR) KN_SHARE_FILE '~1' A GEGEBENFALLS AUS DEM NETZWERK ABMELDEN
A (((13ρ0)≡HABE)∧PROBE∧(0=EBENE))∨(A'v'0#kn_',(ε('ew' 'ws')[1+WS]),'_stat[;9]))/WS KN_SAVE_RESULTS 1' A GEGEBENFALLS ERGEBNISSE
∇ SPEICHERN
A (~PROBE)/'ALX←(2++1>+\\'♦''=ALX)↓ALX' A NOTSPEICHERUNG ENTFERNEN

CharacterRepresentation von KN_STELLUNG_BEWERTEN:

ERG←STEUER KN_STELLUNG_BEWERTEN HABE;WS;SPIEL;VIS;AI;ERSTWURF;GEWICHT;TEILWURF;WERT;V;VV;I
A
A ERMITTELT UND BEWERTET DIE OPTIMALE STRATEGIE FÜR DEN NÄCHSTEN
A SPIELZUG (AUSGEHEND VON DER MOMENTANEN SPIELSTELLUNG)
A
A INITIALISIEREN
A
(WS SPIEL)←2↑A∈('STEUER' '0 0')[1+2#NC 'STEUER'] A STANDARD: ERWARTUNGSWERTE FÜR EINEN EINZELNEN ZUG BERECHNEN
kn_ct←A∈('kn_ct' '0 0')[1+2#NC 'kn_ct'] A GLOBALER BEARBEITUNGSZEIT-MESSER VORHANDEN?
kn_gui←A∈('kn_gui' '4=NC 'AV',')[1+2#NC 'kn_gui'] A STANDARD-UMGEBUNG IN ABHÄNGIGKEIT VOM AV
VIS←A∈('0' '1=wi' 'visible',')[1+kn_gui] A HAUPTFORM SICHTBAR?
AI←[AI[2+~kn_gui] A BEARBEITUNGS-BEGINN NOTIEREN
A (~kn_gui)/'DIE SPIELSTELLUNG '',(A'HABE)', 'WIRD BEWERTET',
A 'ERSTWURF←',(ε5/c'((6f''),'(1 0ρ0'),(ε(c'),(6'),(A'15),(c')p16))) A ALLE MÖGLICHEN ERSTWURFE
(ERSTWURF GEWICHT)←KN_WURF GEWICHTEN ERSTWURF A MÖGLICHE ERSTWURFE GEWICHTEN
ERSTWURF←ERSTWURF[I←A GEWICHT;]◊GEWICHT←GEWICHT[I] A AUFSTIEGEND NACH GEWICHT SORTIEREN (EIGENTLICH UNNOETIG)
TEILWURF←0ρ<0ρ0◊WERT←(0,(13+SPIEL))ρ0
A
A NACH GEWICHTUNG DER MÖGLICHEN ERSTWURFE GETRENNT BEWERTEN

```



```

A
V←((⍲GEWICHT)÷GEWICHT⍲GEWICHT)/GEWICHT A GRUPPEN VON ERSTWURFEN NACH GEWICHT ZUSAMMENFASSEN
⍲(kn_gui↗VIS)/⍲wi ''f_kn_meldung.pb_kn_progress2.value'' (0,(⍲GEWICHT),1 0)' A FORTSCHRITT ANZEIGEN
⍲(kn_gui↗VIS^(~0 0≡kn_ct))/⍲wi ''f_kn_meldung.l_kn_progress3.caption'' (''Geschaetzte Spielstellungs-Bewertungszeit:'' ,⍲AV[14] , (0
  V 1⍲÷ϕkn_ct),'' Sekunden'')
⍲(kn_gui↗VIS^(0 0≡kn_ct))/⍲wi ''f_kn_meldung.l_kn_progress3.caption'' (''Geschaetzte Spielstellungs-Bewertungszeit:'' ,⍲AV[14] ,''
  V nicht ermittelbar'')
I←1
L01:
→(0≡(ERG←HABE WS SPIEL I KN_ERSTWURF_BEWERTEN(V[I]=GEWICHT)≠ERSTWURF))/O A JEWEILS EINE GRUPPE (NORMIERT) BEWERTEN
TEILWURF←TEILWURF,(1≡ERG)⍲WERT←WERT,[1](2≡ERG) A GEWICHTUNG NOCH NICHT BERUECKSICHTIGEN
⍲(kn_gui↗VIS)/⍲wi ''f_kn_meldung.pb_kn_progress2.value'' (+/(GEWICHT⍲V[I])) ⍲ VV←⍲DL .001' A FORTSCHRITT ANZEIGEN
I←I+1
→(I≤⍲V)/L01
kn_ct←kn_ct+(1,⍲AI[2+~kn_gui]-AI)) A BEARBEITUNGS-ZEIT GLOBAL NOTIEREN
ERG←ERSTWURF GEWICHT TEILWURF WERT A ERGEBNISSE UEBERGEHEN (WERT IST UNGEWICHTET)

→(kn_gui)/O
WERT←(GEWICHT×[1]WERT)÷(+/(GEWICHT) A WERTE GEWICHTEN UND NORMIEREN
'WERT DES ZUGES (VERSCHIEDENE ZIELE) ',(⍲+/[1]WERT[;13])
⍲(SPIEL)/''DER WERT DES RESTLICHEN SPIELES BETRAEGT '' , (⍲+/WERT[;14])'
'FUER DIE BEWERTUNG WURDEN ',(⍲⍲AI[2+~kn_gui]-AI), ' SEKUNDEN BENOETIGT'

CharacterRepresentation von KN_ERSTWURF_BEWERTEN:
ERG←STEUER KN_ERSTWURF_BEWERTEN ERSTWURF;HABE;WS;SPIEL;ART;TEILWURF;WURF;GEWICHT;WERT;V;I
A
A BEWERTET ALLE MOEGELICHEN WEITERFUEHRUNGEN EINES ODER MEHRERER
A KNIFFEL-ERSTWUERFE ANHAND DER MOMENTANEN SPIELSTELLUNG
A
A INITIALISIEREN
A
STEUER←⍲∈('STEUER' ,13⍲0')[1+2≠⍲NC 'STEUER'] A STANDARD: ERSTER SPIELZUG
⍲(1≡STEUER)/'STEUER←STEUER 0' A STANDARD: ERWARTUNGSWERTE
(HABE WS SPIEL ART)←4+STEUER,0 0 A FUER EINEN SPIELZUG MAXIMIEREN UND BERECHNEN, KEIN BESTIMMTER ERSTWURF
I←~1⍲ERSTWURF
TEILWURF←(I,0)⍲⍲⍲⍲WERT←(I,0,(13+SPIEL))⍲0
kn_gui←⍲∈('kn_gui' ,4=⍲NC ''AV'') [1+2≠⍲NC 'kn_gui'] A STANDARD-UMGEBUNG IN ABHAENGIGKEIT VOM AV
A
A NACH ANZAHL DER BEIZUBEHALTEN DEN WUERFEL GETRENNT BEWERTEN

```

```

A
I←0
L01:
  WURF←((~1+pERSTWURF),(1 5 10 5 1)[1+I],I)pERSTWURF[';'),(ε('0p0','t5','t1 2 1 3 1 4 1 5 2 3 2 4 2 5 3 4 3 5
  4 5' '1 2 3 1 2 4 1 2 5 1 3 4 1 3 5 1 4 5 2 3 4 5' '1 2 3 4 1 2 3 5 1 2 4 5 1 3 4 5' 't5')[1+I]),']' A
  I WURFEL BEHALTEN
  WURF←1+(KN_WURF_GEWICHTEN WURF) A DIE ZUGEHÖRIGEN TEILWURFE ANORDNEN,
  TEILWURF←TEILWURF,[ppTEILWURF]<[ppWURF]WURF) A NOTIEREN
  2(2=NC 'ART',(εART),(εI))/'(WURF GEWICHT)←ART',(εART),(εI),' ♦ →L02'
  WURF←[0.1+~1+ppWURF]WURF A UND
  WURF←,(ε(5~-1+pWURF)/c'((6/[~1+ppWURF]),'WURF',(ε(c'),[ppWURF]((~2+pWURF),(6'),"(ε"t5~-1+pWURF,"(c'))pt6))) A ERGAENZEN
  (WURF GEWICHT)←KN_WURF_GEWICHTEN WURF A WURFE GEWICHTEN
  0(0≠ART)/'ART',(εART),(εI),'←WURF GEWICHT' A GEBENFALLS RESULTAT GLOBAL SPEICHERN
L02:
  0(0≡(ERG←(HABE GEWICHT WS SPIEL)KN_WURF_BEWERTEN WURF))/0 A GEWICHTET UND NORMIERT BEWERTEN
  WERT←WERT,[~1+ppWERT]ERG A TEILRESULTAT ANHAENGEN
  I←I+1
  →(I≤5)/L01
  I←(1=+I)^(I+V= (~1+pV)/[ppV],[0.1+~1+ppV]/(V+~1/WERT)) A (ERSTE) OPTIMALE STRATEGIE FÜR JEDEN ERSTWURF ERMITTELN
  TEILWURF←',(ε(1=ppI)/'2er','3er','4er','5er','6er','Dreierpass','Vierpass','full house','kleine Strasse','grosse Strasse','Kniffel',
  (ε((0≠HABE)/'1er','2er','3er','4er','5er','6er','Dreierpass','Vierpass','full house','kleine Strasse','grosse Strasse','Kniffel',
  V 'Chance'),"c',''),((^(0=HABE))/'noch keins!'))
  DIE OPTIMALE STRATEGIE DARIN',(εTEILWURF),' ZU BEHALTEN. DIE ERWARTETE PUNKTZAHL BETRÄGT DANN',(ε+/WERT[t13]),', FOLGENDERMASS
  EN VERTEILT AUF DIE VERSCHIEDENEN ZIELE :',(εWERT[t13])
  (SPIEL)''DAS RESTLICHE SPIEL BRINGT WAHRSCHEINLICH NOCH',(εWERT[14]),'' PUNKTE''

CharacterRepresentation von KN_WURF_BEWERTEN:

WERT←STEUER KN_WURF_BEWERTEN WURF:HABE;GEWICHT;WS;SPIEL;PATH;TREFFER;FOLGE;V;I
A
A BEWERTET EINE ODER MEHRERE LISTEN VON KNIFFEL-WUERFEN ANHAND DES
A MOMENTANEN SPIELZUSTANDES
A
A INITIALISIEREN
A

```

```

⊡ WERT[',((~1+ρρWERT)/';'),'14]←(~1+ρFOLGE)ρ(,TREFFER)/(,FOLGE)' ⌈ WERT DER GEWÄHLTEN FOLGE-STELLUNG
⊡ ((2=ρρWERT)^(1=(ρWERT)[~1+ρρWERT])^(14≠1))/'WERT[',((~1+ρρWERT)/';'),'1]←WERT[',((~1+ρρWERT)/';'),'1]+(~1×^/0=WERT[',((~1+ρρWERT)
  ∇ /';'),'13]}' ⌈ ZIEL STREICHEN?
WERT←(0≠V)×(+/[~1+ρρWERT][14/[1+ρρGEWICHT], [0.1+ρρGEWICHT]×WERT)÷(V←14/[ρρGEWICHT], [0.1+~1+ρρGEWICHT]+/[ρρGEWICHT]GEWICHT)
  ∇ ⌈ WERTE GEWICHTET UND NORMIERT SUMMIEREN
⊡ (~SPIEL)/'WERT←WERT[',((~1+ρρWERT)/';'),'13]}' ⌈ SPIEL-RESTWERT VERWERFEN?
→(kn_gui v(1<ρρWERT))/0 ⌈ BEI EINZELLISTE GEGEBENFALLS BOTSCHAFT
'DIE DURCH DIE ANGELEGENE LISTE ERZIELTEN ',(ε('PUNKTE VERTEILEN SICH FOLGENDERMASSEN AUF DIE 13 ZIELE' 'ZIELE WERDEN MIT FOLGENDE
  ∇ N WAHRSCHEINLICHKEITEN ERREICHT')[1+WSI]'),'
⌈ WERT[13]
⊡ (SPIEL)/''DURCH DIE RESULTIERENDE SPIELSTELLUNGEN WERDEN DURCHSCHNITTLLICH WEITERE ',(⌈WERT[14]),'' PUNKTE ERWARTET'',

```

CharacterRepresentation von KN_WURF_GEWICHTEN:

```

ERG←KN_WURF_GEWICHTEN WURF;TREFFER;GEWICHT;I;II;III
⌈
⌈ A SORTIERT UND GEWICHTET EINE ODER MEHRERE LISTEN VON KNIFFEL-WUERFEN
⌈
kn_gui←⊡ε('kn_gui' '4=⊡NC ''⊡AV'',')[1+2≠⊡NC 'kn_gui'] ⌈ STANDARD-UMGEBUNG IN ABHÄNGIGKEIT VOM AV
I←ρWURF ⌈ STRUKTUR NOTIEREN
WURF←,[~1+ρI]WURF ⌈ ALLE WUERFE ZU EINER LISTE ZUSAMMENFASSEN
TREFFER←+/[2]WURF o.=(16) ⌈ WUERFEL NACH AUGENZAHL GRUPPIEREN
WURF←(ρWURF)ρ(,WURF)[⊡((~1+ρWURF)/1×/~1+I),(~1+I)-,TREFFER][((~1+I)/6×~1+1×/~1+I)+,WURF],[1.1](,WURF)] ⌈ JEDEN WURF GRUPPIEREN UN
  ∇ D SORTIEREN
WURF←WURF[III←⊡(II←I[~1+ρI]/1×/~2+I),WURF;] ⌈ WUERFE JEDER LISTE
TREFFER←TREFFER[III;] ⌈ A UND ZUGEHÖRIGE TREFFER NATUERLICH ANORDNEN
WURF←II,WURF[⊡(⊡I),(,[(/TREFFER],[1.1](,+(2=TREFFER));] ⌈ WUERFE JEDER LISTE NACH DEREN STRUKTUR ANORDNEN, LISTEN-ZUGEHÖRIGKEIT
  ∇ NOTIEREN
II←1,1+((v/(WURF≠~1⊡[1]WURF))/1+IρWURF) ⌈ GLEICHARTIGE WUERFE GRUPPIEREN,
WURF←WURF[II;] ⌈ BEIBEHALTEN
GEWICHT←(Λ/0≠WURF)×((1+II,(1+×/~1+I))-II) ⌈ A UND GEWICHTEN
II←WURF[1]1/[WURF[;1] ⌈ A ZU JEDER LISTE GEGEHÖRENDE WUERFE ZAEHLEN
II←(III o.≥(1/[III]))×(II o.+(~1+1/[III←(1+II,(1+ρWURF[;1]))-II))) ⌈ A UND INDIZIEREN
WURF←((~2+I),(/[III),(~1+I)ρ(0,[1]WURF)[1+II;1+~1+I] ⌈ WUERFE
GEWICHT←((~2+I),(/[III)ρ(0,GEWICHT)[1+II] ⌈ A UND GEWICHTUNGEN IN DIE URSPRUENGLICHEN LISTEN UNTERTEILEN
ERG←WURF GEWICHT ⌈ ERGEBNIS UEBERGEHEN

```

APL-Journal

28. Jg. 2009, ISSN 1438-4531

Herausgeber: Rainer Nußbaum,
APL-Germany e.V., Mannheim,
<http://www.apl-germany.de>

Redaktion: Dipl.-Volksw. Martin
Barghoorn (verantw.), TU Berlin,
Franklinstr. 28, D-10587 Berlin,
Tel. (030) 314 24392, Fax (030)
314 25901

Verlag: RHOMBOS-VERLAG, Ber-
lin, Kurfürstenstr. 17, D-10785 Ber-
lin, Tel. (030) 261 9461, Fax (030)
261 6300, eMail: verlag@rhombos.de,
Internet: www.rhombos.de

Erscheinungsweise: halbjährlich

Erscheinungsort: Berlin

Gestaltung: RHOMBOS-VERLAG

Druck: dbusiness.de GmbH, Berlin

Copyright: APL Germany e.V. (für
alle Beiträge, die als Erstveröffent-
lichung erscheinen)

Die Wiedergabe von Gebrauchsnamen,
Handelsnamen, Warenbezeichnungen
usw. in diesem Werk berechtigt auch
ohne besondere Kennzeichnung nicht
zu der Annahme, dass solche Namen im
Sinne der Warenzeichen- und Marken-
schutzgesetzgebung als frei zu betrach-
ten wären und daher von jedermann
benutzt werden dürfen. Eine Haftung
für die Richtigkeit der veröffentlichten
Informationen kann trotz sorgfältiger
Prüfung von Herausgeber und Verlag
nicht übernommen werden.

Mit Namen gekennzeichnete Artikel
geben nicht unbedingt die Meinung
des Herausgebers oder der Redaktion
wieder. Für unverlangte Einsendungen
wird keine Haftung übernommen.
Nachdruck ist nur mit Zustimmung des
Herausgebers sowie mit Quellenangabe
und Einsendung eines Beleges gestat-
tet. Überarbeitungen eingesandter
Manuskripte liegen im Ermessen der
Redaktion.



Allgemeine Informationen

(Stand 2009)

APL-Germany e.V. ist ein gemeinnützi-
ger Verein mit Sitz in Düsseldorf. Sein
Zweck ist es, die Programmiersprache
APL, sowie die Verbreitung des Ver-
ständnisses der Mensch-Maschine
Kommunikation zu fördern. Für In-
teressenten, die zum Gedankenaus-
tausch den Kontakt zu anderen APL-
Benutzern suchen, sowie für solche,
die sich aktiv an der Weiterverbrei-
tung der Sprache APL beteiligen

wollen, bietet APL-Germany den ad-
äquaten organisatorischen Rahmen.

Auf Antrag, über den der Vorstand
entscheidet, kann jede natürliche
oder juristische Person Mitglied wer-
den. Organe des Vereins sind die
mindestens einmal jährlich stattfin-
dende Mitgliederversammlung sowie
der jeweils auf zwei Jahre gewählte
Vorstand.

1. Vorstandsvorsitzender

Dr. Reiner Nussbaum

Dr. Nussbaum gift mbH, Buche-
nerstr. 78, 69259 Mannheim, Tel.
(0621) 7152190.

2. Vorstandsvorsitzender:

Martin Barghoorn

Sekr. FR 6-9, Technische Univer-
sität Berlin, Franklinstr. 28, 10587
Berlin, Tel. (030) 314 24392

Schriftführer

Michael Baas

Dynamic Logistics Systems
GmbH, Wilhelm-Schöffer-Str. 29,
63571 Gelnhausen, Tel. (06051)
13068, Fax (06051) 1642, eMail:
m.baas@dls-planung.de

Schatzmeister

Uwe Janza

Janza Software Consulting, Bau-
erngasse 18, 90443 Nürnberg,
Tel. 0911-327075, Fax 0911-
327076

Beitragssätze:

Ordentliche Mitglieder:

Natürliche Personen 32,- EUR*

Studenten / Schüler 11,- EUR*

Außerordentliche

Mitglieder:

Jurist./natürl. Pers. 500,- EUR*

* Jahresbeitrag

Bankverbindung:

BVB Volksbank eG Bad Vilbel

BLZ 518 613 25

Konto-Nr. 523 2694

Hinweis:

Wir bitten alle Mitglieder, uns
Adressenänderungen und neue
Bankverbindungen immer so-
fort mitzuteilen. Geben Sie bei
Überweisungen den Namen und/
oder die Mitgliedsnummer an.



Einzugsermächtigung

Mitglieds-Nr.: _____

Ich erkläre mich hiermit widerruflich damit einverstanden, daß APL Germany e.V. den jeweils gültigen Jahres-Mitgliedsbeitrag von meinem unten angegebenen Konto abbucht. Mit der Datenübermittlung an das oben genannte Kreditinstitut bin ich einverstanden. Einen eventuell bestehenden Dauerauftrag habe ich bei meiner Bank gelöscht.

Bankbezeichnung: _____

BLZ: _____

Konto-Nr.: _____

Datum: _____

Unterschrift: _____

