Thomas Ströhlein's Endgame Tables: a 50th Anniversary

Article

Supplemental Material

TS' doctoral thesis (OCR readable)

Untersuchungen über kombinatorische Spiele

THOMAS STRÖHLEIN

UNTERSUCHUNGEN ÜBER KOMBINATORISCHE SPIELE

Von der
Fakultät für Allgemeine Wissenschaften
der Technischen Hochschule München
zur Erlangung des akademischen Grades
eines Dr. rer. nat. genehmigte Dissertation

Vorgelegt von
Diplom-Mathematiker
Thomas Ströhlein
geboren zu Neuendettelsau

1. Berichterstatter: o. Prof. Dr. F.L. Bauer
2. Berichterstatter: o. Prof. Dr. G. Aumann

Tag der Einreichung der Arbeit: 22. Januar 197o
Tag der Annahme der Arbeit: 4. Februar 197o
Tag der Promotion: 23. Februar 197o

Inhaltsübersicht

Einleitung

Die vorliegende Arbeit beschäftigt sich mit einer speziellen Klasse von Spielen, den alternierenden Gewinn-Verlust-Spielen mit vollständiger Information. Schon von Neumann und Morgenstern [8] haben gezeigt, daß Spiele mit vollständiger Information determiniert sind, d.h. daß sie einen Wert haben, der durch eine Strategie gewährleistet wird. Diesen Sachverhalt weisen wir im Spezialfall eines Gewinn-Verlust-Spieles direkt nach. Dabei bemühen wir uns, eine Definition von Gewinn (oder des Wertes) zu geben, die dem Bedürfnis nach Strenge und Anlehnung an die Anschauung nachkommt. Wir verwenden dazu den Begriff der Strategie.

Für die Menge der Gewinnstrategien zeigen wir in §3 wesentliche Eigenschaften, mit denen wir in §5 die Beziehungen zwischen Gewinnstellungen in der Algebra der Booleschen Matrizen formulieren. In §4 charakterisieren wir global-optimale Strategien und stellen eine Beziehung zu lokal-optimalem Verhalten her.

Wir ordnen in §6 jedem Spiel ein Boolesches Gleichungssystem zu, für das wir ein iteratives Lösungsverfahren angeben. Spezielle Boolesche Matrixgleichungen mit expliziten Lösungen werden z.B. bei Luce [6] betrachtet.

Durch den Zusammenhang zwischen der Lösung des Gleichungssystems eines Spieles und der Existenz eines Kernes, erhalten wir die Aussage: Ein zyklisch 2-geteilter Graph hat zwei Kerne, die in dem Sinne ausgezeichnet sind, daß mögliche weitere Kerne ihren Durchschnitt umfassen, aber in ihrer Vereinigung enthalten sind. Nach dem Satz von Richardson [1], der für etwas allgemeinere Graphen als für zyklisch 2-geteilte gilt, ist die Existenz eines Kernes im Spielgraphen gesichert.

In den §§7 und 8 wird ein Algorithmus gewonnen und auf seine Eignung für das Schachspiel untersucht. Für Endspiele, bei denen man an allen Stellungen interessiert ist, konnten die Besonderheiten des Schachs geeignet berücksichtigt werden. Das Verfahren wurde für einige Schachendspiele durchgeführt und so hat sich beispielsweise das Dame gegen Turm-Endspiel als im wesentlichen für die Dame-Partei gewonnen - und zwar in spätestens 31 Zügen - erwiesen.

Herr Prof. Dr. F. L. Bauer hat mich zu dieser Arbeit angeregt. Für wertvolle Hinweise und förderliche Diskussionen bin ich ihm sowie den Herren Dr. G. Schmidt und Dr. L. Zagler zu großem Dank verpflichtet.

§1 Graphen und Boolesche Matrizen

In diesem Paragraphen stellen wir einige Begriffe und vorbereitende Sätze aus dem Gebiet der Verbände und der Graphentheorie zusammen. Eine ausführliche Darstellung und Beweise finden sich bei Birkhoff, Gericke und Berge [2,4,1]. Wir erinnern zunächst an die Definition und Strukturen eines Verbandes.

Unter einem Verband $(X,\vee,\wedge)$ versteht man eine Menge X mit zwei Verknüpfungen $\vee$ und $\wedge$, die beide assoziativ und kommutativ sind, sowie den Absorptionsgesetzen

$$x \wedge (x \vee y) = x \quad , \quad x \vee (x \wedge y) = x$$

genügen. Dann gilt auch die Idempotenz von $\vee$ und $\wedge$:

$$x \wedge x = x = x \vee x \ .$$

In einem Verband wird durch

$$x \leqq y \ :\Leftrightarrow \ x \wedge y = x$$

eine teilweise Ordnung eingeführt. $x \wedge y$ heißt dann das Supremum von x und y und $x \vee y$ das Infimum.

Wir betrachten nur vollständige Verbände, bei denen also auch für X Supremum und Infimum existieren. Bezeichnen wir diese mit I (Universalelement) und 0 (Nullelement), so gilt also

$$0 \wedge x = 0 \ , \ 0 \vee x = x$$

$$I \wedge x = x \ , \ I \vee x = I$$

bzw. $\quad 0 \leqq x \leqq I \quad$ für alle $x \in X$.

Ein Verband heißt distributiv, falls stets die Gleichungen

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

gelten. Diese Distributivgesetze sind auseinander herleitbar.

Ein Verband heißt komplementär, falls jedes Element x ein Komplement - das bei einem distributiven Verband dann sogar eindeutig bestimmt ist und mit $\bar{x}$ bezeichnet wird - besitzt, d.h. es bestehen die Beziehungen

$$x \wedge \bar{x} = 0 \quad \text{und} \quad x \vee \bar{x} = I \ .$$

Die Zuordnung $x \to \bar{x}$ ist ein dualer Automorphismus auf X bzgl. $\vee$ und $\wedge$

$$\overline{x \vee y} = \bar{x} \wedge \bar{y} \quad , \quad \overline{x \wedge y} = \bar{x} \vee \bar{y}$$

mit der involutorischen Eigenschaft

$$\bar{\bar{x}} = x \quad .$$

Ein Verband heißt $\vee$-multiplikativ, falls eine Multiplikation besteht, die mit $\vee$ distributiv ist:

$$x\,(y \vee z) = x\,y \vee x\,z \quad , \quad (y \vee z)\,x = y\,x \vee z\,x \quad .$$

Gibt es ein Element E, das $E\,x = x\,E = x$ für alle $x \in X$ genügt, so bezeichnen wir es als Einselement.

Nach dieser Bereitstellung elementarer Begriffe der Verbandstheorie können wir uns der Definition Boolescher Matrizen zuwenden. Um den Zusammenhang zwischen Matrix-Element und Matrix zu verdeutlichen, bedienen wir uns der Schreibweise:

$$X = \big(X[i,j]\big) \quad .$$

<u>D1.1</u> Eine <u>Boolesche Algebra</u> ist ein komplementärer, distributiver Verband $(B,\vee,\wedge,\bar{\ })$. Die Menge aller n x m -Matrizen mit Koeffizienten in B bezeichnen wir mit $\mathcal{B}_{n,m}$ (Menge der <u>Booleschen Matrizen</u>) und setzen darauf die Operationen aus B fort. Zu $X,Y \in \mathcal{B}_{n,m}$ erklären wir $X \vee Y, X \wedge Y$ und $\bar{X} \in \mathcal{B}_{n,m}$ vermöge

$$\begin{aligned} (X \vee Y)[i,j] &:= X[i,j] \vee Y[i,j] \\ (X \wedge Y)[i,j] &:= X[i,j] \wedge Y[i,j] \\ \bar{X}[i,j] &:= \overline{X[i,j]} \end{aligned}$$

und definieren Matrizen 0,I und E durch

$$O[i,j] := 0 \quad \text{und} \quad I[i,j] := I \quad \text{für alle } i,j$$

$$E[i,j] := \begin{cases} I & \\ & \text{für} \\ 0 & \end{cases} \begin{matrix} i = j \\ \\ i \neq j \end{matrix} \quad .$$

<u>S1.1</u> $(\mathcal{B}_{n,m},\vee,\wedge,\bar{\ })$ ist eine Boolesche Algebra mit Nullelement O und Universalelement I. Für die Relation $\leq$ ergibt sich

$$X \leq Y \iff X[i,j] \leq Y[i,j] \quad \text{für alle } i,j.$$

<u>D1.2</u> Wie üblich führt man die Matrixmultiplikation - als Abbildung $\mathcal{B}_{n,m} \times \mathcal{B}_{m,1} \to \mathcal{B}_{n,1}$ - ein, durch

$$X \cdot Y[i,j] := \bigvee_{k=1}^{m} X[i,k] \wedge Y[k,j] \ ,$$

sowie die Transponierung

$$X^T[i,j] := X[j,i]$$

und das Skalarprodukt für $x \in B$, $X \in \mathfrak{B}_{n,m}$

$$(x \wedge X)[i,j] := x \wedge X[i,j] \ .$$

Da $\vee, \wedge$ assoziativ und distributiv sind, erhält man

<u>S1.2</u> $(\mathfrak{B}_{n,n}, \cdot)$ ist eine Halbgruppe, d.h. die Matrix-Multiplikation ist assoziativ, mit Einselement E.
Für die Transponierung hat man die Formeln

$$(X^T)^T = X \quad , \quad (X\,Y)^T = Y^T\,X^T$$

$$(X \vee Y)^T = X^T \vee Y^T \quad , \quad (X \wedge Y)^T = X^T \wedge Y^T$$

$$X \leq Y \quad \Leftrightarrow \quad X^T \leq Y^T \ .$$

Ferner gilt für alle $X \in \mathfrak{B}_{n,m}$:

$$0 \wedge X = 0\,X = X\,0 = 0 \quad , \quad I \vee X = I \ .$$

<u>S1.3</u> $(\mathfrak{B}_{n,n}, \vee, \wedge, \bar{\ }, \cdot)$ ist ein $\vee$-multiplikativer Verband, d.h. er genügt den Gleichungen

$$X\,(Y \vee Z) = X\,Y \vee X\,Z$$

$$(X \vee Y)\,Z = X\,Z \vee Y\,Z \ .$$

Eine entsprechende Aussage für $\wedge$ ist falsch, wie das Beispiel $I\,(E \wedge \bar{E}) = 0 \neq I = I\,E \wedge I\,\bar{E}$ zeigt. Als Ersatz hat man

<u>S1.4</u> Seien $X,Y,Z \in \mathfrak{B}_{n,n}$. Dann bestehen die Ungleichungen

$$X\,(Y \wedge Z) \leq X\,Y \wedge X\,Z$$

$$(X \wedge Y)\,Z \leq X\,Z \wedge Y\,Z \ .$$

Beweis: Wir weisen nur die erste Ungleichung nach.

$$(X\,Y \wedge X\,Z)[i,j] = (\bigvee_k X[i,k] \wedge Y[k,j]) \wedge (\bigvee_k X[i,k] \wedge Z[k,j])$$

$$= \bigvee_k \bigvee_l X[i,k] \wedge Y[k,j] \wedge X[i,l] \wedge Z[l,j]$$

$$= (\bigvee_k X[i,k] \wedge Y[k,j] \wedge Z[k,j]) \vee (\bigvee_k \bigvee_{l \neq k} X[i,k] \wedge Y[k,j] \wedge X[i,l] \wedge Z[l,j]) \geq (X\,(Y \wedge Z))[i,j] \ .$$

Die Sätze 1 bis 4, die sich alle elementar beweisen lassen, beschreiben in einfacher Weise die Eigenschaften Boolescher Matrizen und bringen etwas Übersicht in eine Vielfalt von Formeln. (Dazu siehe etwa: Give'on und Maghout [5,7]).
Im weiteren bestehe die Boolesche Algebra B nur aus den Elementen O und L (den Wahrheitswerten). Eine Aussage in runden Klammern bedeute stets deren Wahrheitswert.
Wir betrachten nun gerichtete Graphen. Ein Graph $G = (X,\Gamma)$ besteht aus einem Paar von Objekten, nämlich einer endlichen Menge X von Punkten und einer Korrespondenz Γ von X in X, d.h. einer Abbildung von X in die Potenzmenge von X

$$\Gamma : X \rightarrow \mathcal{P}(X) \quad ,$$

die Bögen (Übergänge oder Züge) in X beschreibt. Sei $G = (X,\Gamma)$ mit $X = \{x_1,\ldots,x_n\}$ und $A_G \in \mathcal{L}_{n,n}$. Vermöge

$$A_G[i,j] = L \;\;\Leftrightarrow\;\; x_j \in \Gamma x_i$$

ist die Gesamtheit der Graphen mit n Punkten der Menge der Booleschen Matrizen $\mathcal{L}_{n,n}$ (über $B = \{0,L\}$) eineindeutig zugeordnet. Wir nennen A_G die zu G assoziierte Matrix.
Ein Weg im Graphen ist eine spezielle Folge von Elementen aus X $(x_1,x_2,\ldots)$ mit der Eigenschaft $x_{i+1} \in \Gamma x_i$ für $i=1,2,\ldots$ und ein Zyklus ist ein endlicher Weg $(x_1,\ldots,x_m)$ mit $x_1 = x_m$.

Zu zwei Graphen $G_1 = (X_1,\Gamma_1)$ und $G_2 = (X_2,\Gamma_2)$ kann man den Summengraphen $G_1 + G_2 = (X,\Gamma)$ bilden, gemäß

$$X = X_1 \times X_2 \quad \text{und} \quad \Gamma(x_1,x_2) = \Gamma_1 x_1 \times \{x_2\} \cup \{x_1\} \times \Gamma_2 x_2 \quad .$$

<u>D1.3</u> Ein Graph $G = (X,\Gamma)$ heißt <u>zyklisch 2-geteilt</u>, falls es eine Zerlegung der Punktmenge X in zwei Mengen gibt

$$X = X_1 \dot{\cup} X_2$$

($\dot{\cup}$ disjunkte Vereinigung), so daß

$$\Gamma x \subset X_2 \;\text{ für } x \in X_1 \quad \text{und} \quad \Gamma x \subset X_1 \;\text{ für } x \in X_2 \quad .$$

Gemäß der Zuordnung von Graphen und Booleschen Matrizen, übertragen wir den Begriff der zyklischen 2-Teilung auf Elemente aus $\mathcal{L}_{n,n}$.

<u>S1.5</u> Sei A eine zyklisch 2-geteilte, Boolesche Matrix. Dann gibt es eine Permutation der Zeilen und Spalten, so daß A die Form

$$\begin{pmatrix} O_1 & B_1 \\ B_2 & O_2 \end{pmatrix}$$

hat, wobei O_1, O_2 quadratische Nullmatrizen sind.

<u>D1.4</u> Der <u>Kern</u> eines Graphen (X,Γ) ist eine Menge $K \subset X$ mit den Eigenschaften

Innen-Stabilität: $x \in K \Rightarrow \Gamma x \cap K = \emptyset$

Außen-Stabilität: $x \notin K \Rightarrow \Gamma x \cap K \neq \emptyset$.

Diese Bedingungen sind äquivalent zu

$$x \notin K \Leftrightarrow \Gamma x \cap K \neq \emptyset \ .$$

Falls ein Kern existiert, ist in ihm stets die Menge $\{x \in X \mid \Gamma x = \emptyset\}$ enthalten.

Jede Teilmenge $K \subset X$ kann man eineindeutig durch einen Booleschen Vektor $K^* \in \mathcal{L}_{n,1}$ charakterisieren, etwa durch die Zuordnung

$$K^*[i] = (x_i \notin K) \quad \text{bzw.} \quad K = \{x_i \in X \mid K^*[i] = O\} \ .$$

Wir zeigen nun, wie die Frage nach der Existenz von Kernen eines Graphen $G = (X,\Gamma)$ über seine assoziierte Matrix A_G mit den Lösungen der Booleschen Gleichung

$$X = A_G \overline{X}$$

zusammenhängt. Dieses Ergebnis entspricht einem Kriterium, das Berge [1,Seite 47] mittels charakteristischer Funktionen beschreibt.

<u>S1.6</u> $\qquad K$ Kern von $G \Leftrightarrow K^* = A_G \cdot \overline{K^*}$.

Beweis: Sei K Kern von G, d.h.

$$x_i \notin K \Leftrightarrow \Gamma x_i \cap K \neq \emptyset$$

$$K^*[i] = L \Leftrightarrow (\bigvee_{x_j \in \Gamma x_i} x_j \in K) = L \Leftrightarrow (\bigvee_j A_G[i,j] \wedge \overline{K^*}[j]) = L$$

also $K^* = A_G \overline{K^*}$.

§2 Definition eines alternierenden Gewinn-Verlust-Spieles mit vollständiger Information

Im Anschluß an den vorigen Paragraphen ließe sich die Definition unseres Spieles sehr kurz - nämlich als zyklisch 2-geteilten Graphen - treffen. Um den semantischen Gehalt jedoch hervorzuheben, formulieren wir ausführlich:

D2.1 Für zwei Spieler oder Parteien, die wir mit w (Weiß) und s (Schwarz) bezeichnen, ist ein alternierendes Gewinn-Verlust-Spiel mit vollständiger Information (im Sinne von von Neumann) gegeben durch:

(1) eine endliche Menge P mit N Elementen, die wir Positionen (Stellungen oder Spielzustände) nennen und einem ausgezeichneten Element $p_o \in P$, der Ausgangsstellung;

(2) eine disjunkte Zerlegung von P in nicht-leere Mengen P_w und P_s mit N_w bzw. N_s Elementen, die das Zugrecht für w bzw. s in den Stellungen aus P_w bzw. P_s festlegt

$$P = P_w \dot{\cup} P_s \quad ; \quad N = N_w + N_s \quad \text{und} \quad N_w, N_s \neq o \ ;$$

(3) eine Korrespondenz aus P in P, genannt die Spiel- oder Zugregeln (oder Übergänge)

$$\Gamma : P \longrightarrow \mathcal{P}(P)$$

mit der alternierenden Eigenschaft

$$\Gamma(P_w) \subset \mathcal{P}(P_s)$$
$$\Gamma(P_s) \subset \mathcal{P}(P_w) \quad .$$

Ein Spiel wird also beschrieben durch ein Quadrupel (P_w, P_s, p_o, Γ) mit obigen Eigenschaften. Es bleibt noch das "Ziel" des Spieles festzulegen:

(4) Wir bezeichnen als Verlustpositionen für w die Menge

$$V_w^{(o)} := \{p \in P_w \mid \Gamma p = \emptyset\}$$

und als Verlustpositionen für s die Menge

$$V_s^{(o)} := \{p \in P_s \mid \Gamma p = \emptyset\} .$$

Die Elemente aus $V_w^{(o)} \cup V_s^{(o)}$ werden wir auch o-Verlustpositionen

nennen oder Stellungen, in denen in o Zügen Verlust eintritt. Unter "Spielen" oder Abspiel verstehen wir dabei etwa folgendes: Wegen der disjunkten Zerlegung $P = P_w \dot{\cup} P_s$ steht eindeutig fest, wer in p_o am Zuge ist. Sei dies die Partei w. w wählt eine Position p_1 aus Γp_o und sodann s eine Position p_2 aus Γp_1 usw. Das Spiel endet z.B. mit "Gewinn für w", oder wie wir synonym damit sagen wollen "Verlust für s", falls eine Position aus $V_s^{(o)}$ erreicht ist, in der s also nicht mehr ziehen kann. Dies soll jedoch im nächsten Paragraphen präzisiert und im einzelnen ausgeführt werden.
Das Paar (P,Γ) bezeichnen wir als Spielgraphen, dessen assoziierte Matrix (Spiel- oder Übergangsmatrix) sich wegen der alternierenden Eigenschaft in der Form

$$T = \begin{array}{c} \begin{array}{cc} P_w & P_s \end{array} \\ \left(\begin{array}{c|c} 0 & B_w \\ \hline B_s & 0 \end{array}\right) \end{array} \begin{array}{c} \\ P_w \\ P_s \end{array} \quad \in \mathcal{L}_{N,N}$$

schreiben läßt, wenn dieser Zerlegung eine feste Numerierung $P = P_w \dot{\cup} P_s = \{p_1,\dots,p_{N_w}\} \cup \{p_{N_w+1},\dots,p_N\}$ zugrunde liegt.
Die Übergangsmatrizen für ein Spiel sind also zyklisch 2-geteilte Matrizen.

<u>D2.2</u> Ein Spiel mit dem Graphen (P',Γ') heiße <u>Teilspiel</u> des Spieles mit dem Graphen (P,Γ), falls

1. $P' \subset P$
2. $\Gamma' x = \Gamma x$ für alle $x \in P'$.

Bemerkung: Der Graph eines Teilspieles ist im allgemeinen kein Teilgraph des Spielgraphen im üblichen Sinne.

Wie man sich leicht überlegt, bedeutet die Existenz eines Teilspiels, daß die Matrizen B_w und B_s reduzibel sind.

Zum Abschluß wollen wir einige Überlegungen über die Allgemeinheit unserer Definition eines Spieles, insbesondere die alternierende Bedingung (3) und die Äquivalenz "o-Verlust in p $\Leftrightarrow$ $\Gamma p = \emptyset$" betreffend, anstellen.

a) Betrachten wir ein nicht alternierendes Spiel (P,Γ), so läßt sich ihm durch "Vernachlässigung" der Wege innerhalb P_w bzw. P_s ein alternierendes Spiel (P,Γ^*) zuordnen, das dem Gehalt des ursprünglichen Spieles im wesentlichen entspricht.
Konstruktion von Γ^*: Sei $p_o \in P_w$ (bzw. P_s) und $p' \in P_s$ (bzw. P_w), so erklären wir $p' \in \Gamma^* p_o$ genau dann, wenn gilt
$p' \in \Gamma p_o$ oder
$p' \notin \Gamma p_o$ aber es gibt $p_1,\ldots,p_k \in P_w$ (bzw. P_s) mit $p_{i+1} \in \Gamma p_i$ für $i=o,\ldots,k$ und $p' = p_{k+1}$.

b) Die Forderung "o-Verlust in p $\succ$ $\Gamma p = \emptyset$" erscheint eo ipso sinnvoll und soll nicht weiter untersucht werden.

c) Ist die umgekehrte Bedingung nicht erfüllt, d.h. gibt es $p \in P$ mit $\Gamma p = \emptyset$ und nicht o-Verlust in p, so kann man p etwa als Stellungsremise oder Pattstellung erklären. Zum Begriff des Remis im erweiterten Sinne einer Spielremise siehe D3.2 und D5.1. Durch Einführung neuer Positionen p_w^*, p_s^* und geeigneter Erweiterung von Γ kann man erreichen, daß die Stellungsremisen zu Spielremisen werden und sich ansonsten nichts ändert.

Bezeichnen $P_{ow} \subset P_w$ und $P_{os} \subset P_s$ die Mengen der Pattstellungen, so geschieht dies in naheliegender Weise durch die Festsetzung:

$$P_w^* := P_w \cup \{p_w^*\}\ ,\quad P_s^* := P_s \cup \{p_s^*\}\ ,\quad P^* := P_w^* \cup P_s^*$$
$$P_{ow}^* := P_{ow} \cup \{p_w^*\}\ ,\quad P_{os}^* := P_{os} \cup \{p_s^*\}$$
$$\Gamma^* q = \{p_s^*\} \qquad \text{für } q \in P_{ow}^*$$
$$\Gamma^* q = \{p_w^*\} \qquad \text{für } q \in P_{os}^*$$
$$\Gamma^*|P^* \setminus (P_{ow}^* \cup P_{os}^*) = \Gamma|P \setminus (P_{ow} \cup P_{os}) \ .$$

§3 Normalform eines Spieles und Eigenschaften von Gewinn-Strategie-Mengen

Wir betrachten hier die Gesamtheit $\{(P_w, P_s, p, \Gamma) \mid p \in P\}$ von Spielen, also den Spielgraphen (P,Γ) und führen in enger Anlehnung an die Vorstellungen bei einem "Spiel" einige Begriffe ein.

D3.1 Unter einem Abspiel verstehen wir einen Weg im Spielgraphen, d.h. eine spezielle Folge (endlich oder unendlich) von Positionen

$$\mathfrak{a} = (p_1, p_2, \ldots) \quad \text{mit } p_{i+1} \in \Gamma p_i \quad \text{für } i=1,2,\ldots$$

Die Menge der Abspiele bezeichnen wir mit $\mathfrak{A}$ und führen darin die Operationen ein:

1. Die mengentheoretische Verknüpfung $\cap$ ist für Abspiele erklärt mittels
   $p \in \mathfrak{a} \;:\Leftrightarrow$ es gibt ein i mit $p = p_i$, wobei $\mathfrak{a} = (p_1, p_2, \ldots)$.
2. Länge oder Anzahl der Züge eines Abspiels:
   $|\mathfrak{a}| := \left[\frac{n}{2}\right]$, falls $\mathfrak{a} = (p_1, \ldots, p_n)$
3. Konkatenation von Abspielen $\mathfrak{a} = (p_1, \ldots, p_k)$ und $\mathfrak{b} = (q_1, q_2, \ldots)$ mit $|\mathfrak{a}| < \infty$ und $q_1 \in \Gamma p_k$
   $$\mathfrak{a} \circ \mathfrak{b} := (p_1, \ldots, p_k, q_1, q_2, \ldots).$$
4. $\mathfrak{a} \leq \mathfrak{b} \;:\Leftrightarrow \mathfrak{a} = \mathfrak{b}$ oder $\exists\, \mathfrak{v} \in \mathfrak{A} : |\mathfrak{v}| < \infty$ und $\mathfrak{v} \circ \mathfrak{a} = \mathfrak{b}$
   $\mathfrak{a} \geq \mathfrak{b} \;:\Leftrightarrow \mathfrak{b} \leq \mathfrak{a}$.

Bemerkung: Die Definition der Länge eines Abspieles wurde so getroffen, daß sie die Anzahl der Züge der Partei, die in der Ausgangsstellung am Zuge war, bis zum Ende des Spiels (Gewinn oder Verlust) zählt.
Kann man $\mathfrak{a} \circ \mathfrak{b}$ bilden, so gilt $\mathfrak{a} \circ \mathfrak{b} \in \mathfrak{A}$.

HS3.1 $(\mathfrak{A}, \leq)$ ist eine teilweise geordnete Menge.

Beweis: Sei etwa $\mathfrak{a} \leq \mathfrak{b}$ mit $u_1 \circ \mathfrak{a} = \mathfrak{b}$ und $\mathfrak{b} \leq \mathfrak{c}$ mit $u_2 \circ \mathfrak{b} = \mathfrak{c}$, so gilt $\mathfrak{a} \leq \mathfrak{c}$, da $(u_2 \circ u_1) \circ \mathfrak{a} = \mathfrak{c}$ und alle Bedingungen, die an die Konkatenierbarkeit von $u_2 \circ u_1 \circ \mathfrak{a}$ gestellt werden, erfüllt sind.

D3.2 Sei $\mathfrak{a} \in \mathfrak{A}$ ein Abspiel. Endet diese Folge mit einem p aus der Menge $V_w^{(o)} \cup V_s^{(o)}$ der o-Verluststellungen, oder ist sie eine nicht abbrechende Folge, so sprechen wir von einer Partie und schreiben

$$\mathfrak{a} \in \mathfrak{A}_o \quad .$$

Im Falle $p \in V_s^{(o)}$ heißt die Partie gewonnen für w (= verloren für s), und im Falle $p \in V_w^{(o)}$ heißt die Partie gewonnen für s (= verloren für w), und wir schreiben

$$\mathfrak{a} \in \mathfrak{A}_w \quad \text{bzw.} \quad \mathfrak{a} \in \mathfrak{A}_s \quad .$$

Anderenfalls, d.h. für eine nicht abbrechende Folge, sprechen wir von einer Remis-Partie.

Wir haben also jede Partie "bewertet", etwa in der Form

$$\text{Wert}(\mathfrak{a}) := \begin{cases} 1 & \mathfrak{a} \in \mathfrak{A}_w \\ -1 & \text{falls } \mathfrak{a} \in \mathfrak{A}_s \\ o & \mathfrak{a} \in \mathfrak{A}_o \setminus (\mathfrak{A}_w \cup \mathfrak{A}_s) . \end{cases}$$

Bemerkung: Trivialerweise gilt $\mathfrak{A}_w \cap \mathfrak{A}_s = \emptyset$. Ebenso einsichtig, aber für die weiteren Überlegungen wichtig, ist

HS3.2 Für die Bewertung der Partien, also für die Eigenschaft "Gewinn für w", "Gewinn für s" oder "Remis", gilt

$$\text{Wert}(\mathfrak{a}) = \text{Wert}(\mathfrak{b}) \quad \text{falls } \mathfrak{a} \leq \mathfrak{b} \quad .$$

Bezüglich der Notation der Strategie*) schließen wir an die Arbeit von Gale und Stewart [3] an.

D3.3 Eine Strategie des Spielers w ist eine Funktion

$$\sigma : P_w \setminus V_w^{(o)} \longrightarrow P_s$$

mit der Eigenschaft $\sigma(p) \in \Gamma p$ für alle p.
Die Menge aller w-Strategien bezeichnen wir mit Σ_w .

*) Der Begriff der Strategie wurde von Steinhaus [9], 1925 als wichtiges Hilfsmittel der Spieltheorie erkannt und empfohlen.

Entsprechend ist für den Spieler s die Strategiemenge definiert

$$\Sigma_s := \{ \tau \in P_w^{P_s \setminus V_s^{(0)}} \mid \tau(p) \in \Gamma p \text{ für alle } p \} .$$

Zu vorgegebener Ausgangsstellung p und fest gewählten Strategien $\sigma \in \Sigma_w$ und $\tau \in \Sigma_s$ ist genau eine Partie bestimmt, die wir mit

$$\langle p;\sigma,\tau\rangle := \begin{cases} (p,\sigma(p),\tau\sigma(p),\dots) & \text{falls } p \in P_w \\ (p,\tau(p),\sigma\tau(p),\dots) & \phantom{\text{falls }} p \in P_s \end{cases}$$

bezeichnen wollen.

Falls $p \in V_w^{(0)} \cup V_s^{(0)}$, ergibt die Definition $\langle p;\sigma,\tau\rangle = (p)$. Die Partien der Form $\langle p;\sigma,\tau\rangle$ genügen den elementaren Identitäten

$$\langle p;\sigma,\tau\rangle = \begin{cases} (p) \circ \langle\sigma(p);\sigma,\tau\rangle & \text{falls } p \in P_w \\ (p) \circ \langle\tau(p);\sigma,\tau\rangle & \phantom{\text{falls }} p \in P_s \end{cases} .$$

Unter der Aussage "Spieler w wählt Strategie σ" verstehen wir die a priori-Entscheidung von w, in Position p' den Zug $\sigma(p')$ auszuführen (unabhängig davon, ob w während des Spieles die Position p' überhaupt erreicht).
Die Zuordnung $P \times \Sigma_w \times \Sigma_s \ni (p,\sigma,\tau) \to \langle p;\sigma,\tau\rangle \in \mathfrak{A}_b$ ist natürlich nicht injektiv, da σ auf $\{q \in P_w \mid q \notin \langle p;\sigma,\tau\rangle\}$ und τ auf $\{q \in P_s \mid q \notin \langle p;\sigma,\tau\rangle\}$ beliebig gewählt werden dürfen, ohne die betrachtete Partie zu ändern. Die gesamte Strategiemenge ist also zur Beschreibung der Partien unnötig groß, und jede einzelne Strategie kann unnötige "Angaben" enthalten.

HS3.3 Durch Strategien erzeugte Partien sind spezielle Partien, in dem Sinne, daß ihre Gesamtheit womöglich echt in $\mathfrak{A}_b$ enthalten ist. Betrachten wir nämlich $\alpha = \langle p;\sigma,\tau\rangle = (p_1,p_2,\dots)$ als Weg im Spielgraphen, so gilt für eine Gewinn-Partie, daß die Anzahl der Positionen der Folge $< N$ ist, während eine Remis-Partie durch einen Zyklus der Länge $\leq N$ charakterisiert wird:

(1) Sei $q \in \alpha = \langle p;\sigma,\tau\rangle$. Dann gilt

$$\alpha \in \mathfrak{A}_w \cup \mathfrak{A}_s \quad \Leftrightarrow \quad q = p_i \text{ für genau ein } i$$

Und in diesem Fall ist $|\alpha| \leq \left[\frac{N-1}{2}\right]$.

(2) Sei $p_i = p_j \in \mathfrak{m}$ für $i \neq j$, etwa $i < j$. Dann gilt
$= (p_1, \ldots, p_{i-1}) \circ (p_i, \ldots, p_{j-1})^{\infty} \doteq (p_i, \ldots, p_{j-1})^{\infty} \in \mathfrak{A}_o \setminus (\mathfrak{A}_w \cup \mathfrak{A}_s)$.

Beweis: Es genügt im wesentlichen (2) zu zeigen. Aus $p_i = p_j$ folgt unmittelbar $p_{i+1} = p_{j+1}$ (denn ist etwa $p_i \in P_w$, so ist $p_{i+1} = \sigma(p_i) = \sigma(p_j) = p_{j+1}$) und durch Induktion $p_{i+n} = p_{j+n}$. Die Beachtung von $p_i \in \Gamma p_{j-1}$ führt zur Schreibweise für $\mathfrak{m}$ in (2).

Mit der Bewertung der Abspiele aus D3.2 haben wir für jedes p_o die bekannte Matrixform (Normalform) des Spieles erhalten: Sei $\Sigma_w = \{\sigma_1, \ldots, \sigma_n\}$ und $\Sigma_s = \{\tau_1, \ldots, \tau_m\}$, so ist die Spielmatrix M definiert durch

$$M[i,j] := \mathrm{Wert}(\langle p_o; \sigma_i, \tau_j \rangle) \quad .$$

Die Menge der Strategien ist sehr umfangreich - im Extremfall ist $|\Sigma_w| = N^{N_w}$, $|\Sigma_s| = N^{N_s}$ und M hat N^N Elemente - und wird von uns im wesentlichen nur zur Gewinnung allgemeiner Sätze verwendet. Unserer Definition angemessen, betrachten wir Spiele später in einer extensiven Form, bei der die einzelnen Züge im Vordergrund stehen.

Wir wollen für jede Stellung Gewinn (oder Verlust) für w bzw. s erklären. Ausgehend von den bereits definierten o-Verluststellungen haben wir schon für Partien Gewinn erklärt und werden dies auf Strategien und dadurch schließlich auf Stellungen fortsetzen. Zunächst definieren wir also Gewinn für Strategien und bezeichnen eine Strategie $\sigma \in \Sigma_w$ dann als Gewinnstrategie für w, falls sich bei jeder Wahl der s-Strategie eine für w gewonnene Partie ergibt.

<u>D3.4 Gewinnstrategie-Mengen</u> für w bzw. s in $p \in P$:

$$\Sigma_w(p) := \{\sigma \in \Sigma_w \mid \langle p; \sigma, \tau \rangle \in \mathfrak{A}_w \text{ für alle } \tau \in \Sigma_s\}$$
$$\Sigma_s(p) := \{\tau \in \Sigma_s \mid \langle p; \sigma, \tau \rangle \in \mathfrak{A}_s \text{ für alle } \sigma \in \Sigma_w\}$$

Bemerkung: $\Sigma_w(p) \neq \emptyset \succ \Sigma_s(p) = \emptyset$
$\Sigma_s(p) \neq \emptyset \succ \Sigma_w(p) = \emptyset$.

Dies ergibt sich unmittelbar aus der Bemerkung zu D3.2 $\mathfrak{A}_w \cap \mathfrak{A}_s = \emptyset$. Die Annahme $\Sigma_w(p) \neq \emptyset$ und $\Sigma_s(p) \neq \emptyset$ impliziert die Existenz von Elementen $\sigma' \in \Sigma_w$ und $\tau' \in \Sigma_s$ mit $\langle p;\sigma',\tau\rangle \in \mathfrak{A}_w$ für alle $\tau \in \Sigma_s$ und $\langle p;\sigma,\tau'\rangle \in \mathfrak{A}_s$ für alle $\sigma \in \Sigma_w$ und somit $\langle p;\sigma',\tau'\rangle \in \mathfrak{A}_w \cap \mathfrak{A}_s$. Widerspruch!

Falls $p \in V_w^{(o)}$ ergibt sich wegen der Definition von σ, $\Sigma_w(p) = \emptyset$ und wir erhalten:

$$\Sigma_w(p) = \begin{cases} \emptyset & \text{falls } p \in V_w^{(o)} \\ \Sigma_w & \phantom{\text{falls }} p \in V_s^{(o)} \end{cases}$$

$$\Sigma_s(p) = \begin{cases} \emptyset & \text{falls } p \in V_s^{(o)} \\ \Sigma_s & \phantom{\text{falls }} p \in V_w^{(o)} \end{cases} .$$

Für Gewinnstrategien beweisen wir einen wichtigen Satz, der auf HS3.2 beruht:

<u>S3.4</u> Aus $\sigma \in \Sigma_w(p)$ folgt auch für alle $\tau \in \Sigma_s$ und alle $p' \in \langle p;\sigma,\tau\rangle$, daß gilt $\sigma \in \Sigma_w(p')$.

Beweis: $\sigma \in \Sigma_w(p)$ bedeutet

(1) $\langle p;\sigma,\tau\rangle \in \mathfrak{A}_w$ für alle $\tau \in \Sigma_s$.

Wir wählen ein beliebiges $\tau' \in \Sigma_s$ und ein beliebiges

(2) $p' \in \langle p;\sigma,\tau'\rangle$.

Nun nehmen wir an, die Behauptung sei falsch, d.h. $\sigma \notin \Sigma_w(p')$ oder

(3) $\exists\, \tau'' \in \Sigma_s \;:\; \langle p';\sigma,\tau''\rangle \notin \mathfrak{A}_w$.

Aus (1) und (2) ergibt sich mit HS3.3

(4) $\langle p;\sigma,\tau'\rangle = \omega \cdot \langle p';\sigma,\tau'\rangle$ mit $\omega = (p_1,\ldots,p_r)$, wobei $p_1 = p$.

Wir setzen $p_{r+1} := p'$ und betrachten $(p_1,\ldots,p_{r+1}) \in \mathfrak{A}$.

(5a) $p_1 \notin \langle p';\sigma,\tau''\rangle$,
denn sonst wäre $\langle p;\sigma,\tau''\rangle \trianglelefteq \langle p';\sigma,\tau''\rangle$ und dies kann nach HS3.2 nicht sein, da $\langle p;\sigma,\tau''\rangle \in \mathfrak{A}_w$ und $\langle p';\sigma,\tau''\rangle \notin \mathfrak{A}_w$.

(5b) $p_{r+1} \in \langle p';\sigma,\tau''\rangle$.

Aus (5a,b) folgt nun, daß es ein maximales k gibt, so daß gilt

(6) $p_1,\ldots,p_{k-1} \notin \langle p';\sigma,\tau''\rangle$ und $p_k \in \langle p';\sigma,\tau''\rangle$.

Wegen $\langle p_k;\sigma,\tau''\rangle \trianglelefteq \langle p';\sigma,\tau''\rangle$ ergibt sich dann sogar

(6)' $p_1,\dots,p_{k-1} \notin \langle p_k;\sigma,\tau''\rangle$ und $\langle p_k;\sigma,\tau''\rangle \notin \mathfrak{A}_w$.

Nun definieren wir eine Strategie $\tau_o \in \Sigma_s$ durch

$$\tau_o(q) := \begin{cases} \tau'(q) & \text{für } q \notin \langle p_k;\sigma,\tau''\rangle \\ \tau''(q) & \text{sonst} \end{cases} .$$

Wegen (6)' ergibt sich für die Partie $\langle p_1;\sigma,\tau_o\rangle$ zunächst eine Anfangsfolge von $\langle p;\sigma,\tau'\rangle$ und sodann $\langle p_k;\sigma,\tau''\rangle$, also

(7) $\langle p;\sigma,\tau_o\rangle = (p_1,\dots,p_{k-1})\circ\langle p_k;\sigma,\tau''\rangle$

und weiter erhält man nun $\langle p;\sigma,\tau_o\rangle \notin \mathfrak{A}_w$. Dies steht im Widerspruch zu (1).

Mit Hilfe dieses Satzes zeigen wir zunächst noch eine bemerkenswerte Eigenschaft von Gewinnstrategie-Mengen, die im weiteren Verlauf folgendes ergibt: Die Gewinnstrategie-Menge in p_1 ist, falls sie nicht leer ist, dann sogar so groß, daß es darin Strategien gibt, die auch in jeder anderen Position p_2 mit $\Sigma_w(p_2) \neq \emptyset$ gewinnen.

<u>S3.5</u> Sei $p_1,p_2 \in P$ mit $\Sigma_w(p_1) \neq \emptyset$ und $\Sigma_w(p_2) \neq \emptyset$.

Dann gilt sogar $\Sigma_w(p_1) \cap \Sigma_w(p_2) \neq \emptyset$.

Bemerkung: Es gilt $\bigcap\limits_{p \in P:\Sigma_w(p)\neq\emptyset} \Sigma_w(p) \neq \emptyset$.

Beweis: Sei $\sigma_1 \in \Sigma_w(p_1)$ und $\sigma_2 \in \Sigma_w(p_2)$. Wir wollen σ_2 so zu einem σ_o abändern, daß gilt $\sigma_o \in \Sigma_w(p_1) \cap \Sigma_w(p_2)$. Wir betrachten die Menge

$$H := \{p \in P_w \mid p \in \langle p_2;\sigma_2,\tau\rangle \text{ für mindestens ein } \tau \in \Sigma_s\}$$

also alle Positionen, die durch σ_2 ausgehend von p_2 erreicht werden können. Nach S3.4 ergibt sich

(1) $p \in H \;\succ\; \sigma_2 \in \Sigma_w(p)$

Mittels H definieren wir eine Strategie $\sigma_o \in \Sigma_w$

$$\sigma_o(q) := \begin{cases} \sigma_1(q) & \text{für } q \notin H \\ \sigma_2(q) & \text{für } q \in H \end{cases} ,$$

die folgende Eigenschaft besitzt: Eine Partie, die seitens w mit σ_0 geführt wird und einmal eine Stellung aus H erreicht hat, bleibt stets in H, folgt also dann der Strategie σ_2, d.h.

(2) $p \in P$ und $\tau \in \Sigma_s$ beliebig; $p' \in \langle p;\sigma_0,\tau\rangle \cap H$

$$\succ \quad \langle p';\sigma_0,\tau\rangle = \langle p';\sigma_2,\tau\rangle$$

Wenn wir zeigen können

$$p'' := \tau\sigma_2(p') \in H$$

so ergibt sich daraus (2) durch Induktion.

Zu $p' \in H$ hat man nach Definition ein τ', so daß $p' \in \langle p_2;\sigma_2,\tau'\rangle$.

Damit erklären wir ein $\tau'' \in \Sigma_s$

$$\tau''(q) := \begin{cases} \tau'(q) & \text{für } q \in \langle p_2;\sigma_2,\tau'\rangle, \text{ aber } q \notin \langle p';\sigma_2,\tau'\rangle \\ p'' & \text{für } q = \sigma_2(p') \\ \text{beliebig} & \text{sonst} \end{cases}$$

wofür gilt $p'' \in \langle p_2;\sigma_2,\tau''\rangle$ d.h. $p'' \in H$. Und (2) ist bewiesen.

Da $p_2 \in H$, ergibt sich mittels (2) für beliebiges $\tau \in \Sigma_s$

$$\langle p_2;\sigma_0,\tau\rangle = \langle p_2;\sigma_2,\tau\rangle \qquad \text{und somit}$$

$$\sigma_0 \in \Sigma_w(p_2) \ .$$

Zum Nachweis $\sigma_0 \in \Sigma_w(p_1)$ treffen wir eine Fallunterscheidung; dabei sei stets $\tau \in \Sigma_s$ beliebig, und wir verwenden (2).

1.Fall: $p_1 \in H$ $\qquad \langle p_1;\sigma_0,\tau\rangle = \langle p_1;\sigma_2,\tau\rangle$

2.Fall: $p_1 \notin H$

$$\langle p_1;\sigma_0,\tau\rangle = \begin{cases} \langle p_1;\sigma_1,\tau\rangle \\ w \circ \langle p^*;\sigma_2,\tau\rangle \end{cases} \text{falls } \langle p_1;\sigma_1,\tau\rangle \cap H \begin{cases} = \emptyset \\ \neq \emptyset \end{cases}$$

Dabei ist $p^* \in H$ wie folgt bestimmt. Schreiben wir $\langle p_1;\sigma_1,\tau\rangle = (q_1,q_2,\ldots)$ und setzen $q_{r+1} = p^*$ so gilt

$$w = (q_1,\ldots,q_r) \quad \text{und} \quad w \cap H = \emptyset \ .$$

Also ist in diesem Falle entweder $\langle p_1;\sigma_0,\tau\rangle = \langle p_1;\sigma_1,\tau\rangle$ oder $\langle p_1;\sigma_0,\tau\rangle \hat{=} \langle p^*;\sigma_2,\tau\rangle$ und somit

$$\sigma_0 \in \Sigma_w(p_1) \ .$$

<u>S3.6</u> Für die Gewinnstrategien in benachbarten Positionen besteht ein enger Zusammenhang:

(w1) Für $p \in P_w$ gilt

$$\Sigma_w(p) \subset \bigcup_{p' \in \Gamma p} \Sigma_w(p') \quad ,$$

sowie $\quad \Sigma_w(p) \neq \emptyset \quad \Leftrightarrow \quad \bigcup_{p' \in \Gamma p} \Sigma_w(p') \neq \emptyset .$

(w2) Für $p \in P_s$ gilt

$$\Sigma_w(p) = \bigcap_{p' \in \Gamma p} \Sigma_w(p') \, ,$$

sowie $\quad \Sigma_w(p) \neq \emptyset \quad \Leftrightarrow \quad \Sigma_w(p') \neq \emptyset$ für alle $p' \in \Gamma p$.

(s1) Für $p \in P_s$ gilt

$$\Sigma_s(p) \subset \bigcup_{p' \in \Gamma p} \Sigma_s(p') \, ,$$

sowie $\quad \Sigma_s(p) \neq \emptyset \quad \Leftrightarrow \quad \bigcup_{p' \in \Gamma p} \Sigma_s(p') \neq \emptyset .$

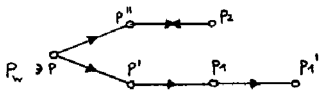
(s2) Für $p \in P_w$ gilt

$$\Sigma_s(p) = \bigcap_{p' \in \Gamma p} \Sigma_s(p') \, ,$$

sowie $\quad \Sigma_s(p) \neq \emptyset \quad \Leftrightarrow \quad \Sigma_s(p') \neq \emptyset$ für alle $p' \in \Gamma p$.

Bemerkungen: Die Menge der Gewinnstrategien für w in Stellungen aus P_s ist durch diejenigen in Stellungen aus P_w bestimmt; entsprechend für s.
Mit der Konvention $\bigcup_{\emptyset} = \emptyset$ und $\bigcap_{\emptyset} = \Sigma_w$ gelten die Formeln auch für $p \in V_w^{(o)} \cup V_s^{(o)}$, d.h. $\Gamma p = \emptyset$.

Die Inklusion " $\supset$ " in (w1) bzw. (s1) muß nicht bestehen, wie das einfache (als Spielgraph gezeichnete) Gegenbeispiel zeigt:



$$V_s^{(o)} = \{p_1'\}$$
$$\Sigma_w(p') = \Sigma_w$$
$$\sigma \in \Sigma_w(p) \Leftrightarrow \sigma(p) = p'$$

Und somit gilt für ein σ^* das $\sigma^*(p) = p''$ genügt $\sigma^* \in \Sigma_w(p')$ aber $\sigma^* \notin \Sigma_w(p)$.

Beweis des S3.6 o.E. für w-Strategien:

ad (w1): Wir beweisen " $\subset$ " und " $\Leftarrow$ "; dazu genügt es zu zeigen

$$\sigma \in \Sigma_w(p) \quad \Leftrightarrow \quad \sigma \in \Sigma_w(\sigma(p)) \; .$$

Sei $\sigma \in \Sigma_w(p)$. Da $p \in P_w$ ist stets (für beliebiges $\tau \in \Sigma_s$) $\sigma(p) \in \langle p;\sigma,\tau \rangle$ und somit nach S3.4 $\sigma \in \Sigma_w(p')$ für $p' = \sigma(p)$.

Sei nun umgekehrt $\sigma' \in \Sigma_w(p')$ für $p' \in \Gamma p$.

Fall a: $p \in \langle p';\sigma',\tau_1\rangle$ für ein $\tau_1 \in \Sigma_s$.

Dann ist nach S3.4 auch $\sigma' \in \Sigma_w(p)$.

Fall b: $p \notin \langle p';\sigma',\tau\rangle$ für alle $\tau \in \Sigma_s$.

Dann definieren wir ein $\sigma \in \Sigma_w$ durch

$$\sigma(q) := \begin{cases} p' & \text{für } q = p \\ \sigma'(q) & \text{sonst} \end{cases}$$

und erhalten für alle $\tau \in \Sigma_s$ (da $p \in P_w$)

$\langle p;\sigma,\tau\rangle = (p)\circ\langle p';\sigma',\tau\rangle$

und somit $\langle p;\sigma,\tau\rangle \in \mathfrak{A}_w$ für alle $\tau \in \Sigma_s$

d.h. $\sigma \in \Sigma_w(p)$.

ad (w2): Sei $\sigma \in \Sigma_w(p)$. Aus $p \in P_s$ und S3.4 ergibt sich somit

$\sigma \in \Sigma_w(\tau(p))$ für alle $\tau \in \Sigma_s$

und wegen $\{\tau(p) \mid \tau \in \Sigma_s\} = \Gamma p$ auch $\sigma \in \bigcap_{p' \in \Gamma p} \Sigma_w(p')$.

Sei $\sigma \in \bigcap_{p' \in \Gamma p} \Sigma_w(p')$, d.h. für alle $p' \in \Gamma p$ und alle $\tau \in \Sigma_s$

ist $\langle p';\sigma,\tau\rangle \in \mathfrak{A}_w$. Für alle $\tau \in \Sigma_s$ gilt aber auch

$\langle p;\sigma,\tau\rangle = (p)\circ\langle \tau(p);\sigma,\tau\rangle$ und somit wegen $p' = \tau(p)$

$\langle p;\sigma,\tau\rangle \in \mathfrak{A}_w$ d.h. $\sigma \in \Sigma_w(p)$.

Die noch zu beweisende Implikation $\Sigma_w(p') \neq \emptyset$ für alle $p' \in \Gamma p$ $\succ \Sigma_w(p) \neq \emptyset$ ergibt sich unmittelbar aus S3.5, da dann

$\bigcap_{p' \in \Gamma p} \Sigma_w(p') \neq \emptyset$. q.e.d.

Im Hinblick auf Fragen der Art "Gewinn in wieviel Zügen", treffen wir anschließend an D3.4 eine Unterteilung der Gewinnstrategie-Mengen.

<u>D3.5</u> Für $n \geq 0$ bezeichnen wir

$$\Sigma_w^{[n]}(p) := \{\sigma \in \Sigma_w(p) \mid |\langle p;\sigma,\tau\rangle| \leq n \text{ für alle } \tau \in \Sigma_s\}$$
$$\Sigma_s^{[n]}(p) := \{\tau \in \Sigma_s(p) \mid |\langle p;\sigma,\tau\rangle| \leq n \text{ für alle } \sigma \in \Sigma_w\}$$

als Menge der Strategien, die bei beliebiger Gegenwehr <u>in höchstens n Zügen zum Gewinn</u> führen.

Einige elementare Eigenschaften dieser Mengen fassen wir in folgendem Hilfssatz zusammen.

HS3.7 Wir formulieren o.E. nur für die w-Strategien:

(1) Die $(\Sigma_w^{[n]}(p))_{n \geq o}$ bilden eine aufsteigende Folge

$$\Sigma_w^{[n-1]}(p) \subset \Sigma_w^{[n]}(p)$$

(2) $$\Sigma_w^{[o]}(p) = \begin{cases} \Sigma_w & \text{für } p \in V_s^{(o)} \\ \emptyset & \text{sonst} \end{cases}$$

(3) $$\bigcup_{n \geq o} \Sigma_w^{[n]}(p) = \Sigma_w(p) \quad .$$

Wegen HS3.3 kann man etwa $n \leq \left[\frac{N-1}{2}\right]$ setzen.

Mit S3.6 erhält man analoge Aussagen für die Unterteilungen der Gewinnstrategie-Mengen.

S3.8 Beziehungen der Strategie-Mengen $\Sigma_w^{[n]}(p)$ bzw. $\Sigma_s^{[n]}(p)$:

(w1) Für $p \in P_w$ gilt

$$\Sigma_w^{[n]}(p) \subset \bigcup_{p' \in \Gamma p} \Sigma_w^{[n-1]}(p') \ ,$$

sowie $\Sigma_w^{[n]}(p) \neq \emptyset \iff \bigcup_{p' \in \Gamma p} \Sigma_w^{[n-1]}(p') \neq \emptyset$.

(w2) Für $p \in P_s$ gilt

$$\Sigma_w^{[n]}(p) = \bigcap_{p' \in \Gamma p} \Sigma_w^{[n]}(p') \ ,$$

sowie $\Sigma_w^{[n]}(p) \neq \emptyset \iff \Sigma_w^{[n]}(p') \neq \emptyset$ für alle $p' \in \Gamma p$.

(s1) Für $p \in P_s$ gilt

$$\Sigma_s^{[n]}(p) \subset \bigcup_{p' \in \Gamma p} \Sigma_s^{[n-1]}(p') \ ,$$

sowie $\Sigma_s^{[n]}(p) \neq \emptyset \iff \bigcup_{p' \in \Gamma p} \Sigma_s^{[n-1]}(p') \neq \emptyset$.

(s2) Für $p \in P_w$ gilt

$$\Sigma_s^{[n]}(p) = \bigcap_{p' \in \Gamma p} \Sigma_s^{[n]}(p') \ ,$$

sowie $\Sigma_s^{[n]}(p) \neq \emptyset \iff \Sigma_s^{[n]}(p') \neq \emptyset$ für alle $p' \in \Gamma p$.

Beim Beweis benützen wir die Gültigkeit des S3.6 und berücksichtigen die Länge der entstehenden Partien.

ad (w1): Sei $\sigma \in \Sigma_w^{[n]}(p)$. Dies ist nach S3.6 äquivalent zu

$$\sigma \in \Sigma_w^{[n]}(p) \cap \Sigma_w(\sigma(p))$$

$$\Leftrightarrow \quad \sigma \in \Sigma_w(p) \cap \Sigma_w(\sigma(p)) \text{ und } |\langle p;\sigma,\tau\rangle| \leq n \quad (\text{für alle } \tau \in \Sigma_s)$$

Wegen $\langle p;\sigma,\tau\rangle = (p)\circ\langle \sigma(p);\sigma,\tau\rangle$ und $|\langle \sigma(p);\sigma,\tau\rangle| \leq n-1$ ist weiter

$$\Leftrightarrow \quad \sigma \in \Sigma_w(p) \cap \Sigma_w^{[n-1]}(\sigma(p))$$

$$\Leftrightarrow \quad \sigma \in \Sigma_w^{[n-1]}(p') \quad \text{für } p' = \sigma(p)$$

$$\Leftrightarrow \quad \bigcup_{p' \in \Gamma p} \Sigma_w^{[n-1]}(p') \neq \emptyset$$

ad (w2): Wir zeigen nur " = ". Unter Beachtung von $|\langle p;\sigma,\tau\rangle| = |\langle \tau(p);\sigma,\tau\rangle|$ und $\{\tau(p) \mid \tau \in \Sigma_s\} = \Gamma p$ ergibt sich die Behauptung gemäß

$$\sigma \in \Sigma_w^{[n]}(p)$$

$$\Leftrightarrow \quad \sigma \in \Sigma_w^{[n]}(p) \cap \bigcap_{p' \in \Gamma p} \Sigma_w(p')$$

$$\Leftrightarrow \quad \sigma \in \bigcap_{p' \in \Gamma p} \Sigma_w(p') \quad \text{und} \quad |\langle p;\sigma,\tau\rangle| \leq n \quad \text{für alle } \tau \in \Sigma_s$$

$$\Leftrightarrow \quad \sigma \in \bigcap_{p' \in \Gamma p} \Sigma_w(p') \quad \text{und} \quad |\langle p';\sigma,\tau\rangle| \leq n \quad \text{ - " - und } p' \in \Gamma p$$

$$\Leftrightarrow \quad \sigma \in \bigcap_{p' \in \Gamma p} \Sigma_w^{[n]}(p') .$$

§4 Charakterisierung von optimalen Strategien

Mit Hilfe der in §3 eingeführten Gewinnstrategien wollen wir optimale Strategien in beliebigen Stellungen erklären. Um von einem optimalen Verhalten einer Partei in einer Stellung sprechen zu können, treffen wir zunächst folgende Definition.

D4.1 Wir erklären eine Funktion $z : P \to \mathbb{N}$.

$$z(p) := \begin{cases} \min\limits_{\Sigma_w^{[n]}(p) \neq \emptyset} \{n\} & \text{Fall 1:} \quad \Sigma_w(p) \neq \emptyset \\ \min\limits_{\Sigma_s^{[n]}(p) \neq \emptyset} \{n\} & \text{Fall 2:} \quad \Sigma_s(p) \neq \emptyset \\ \infty & \text{Fall 3:} \quad \Sigma_w(p) = \Sigma_s(p) = \emptyset \end{cases}$$

heiße Zugzahl der Stellung p.

Vereinbarung: In diesem Paragraphen beziehen wir uns stets auf die Fallunterscheidung dieser Definition.

Mit D4.1 kann man im Fall 1 von $\Sigma_w^{[n]}(p)$ mit $n := z(p)$ als den optimalen Gewinnstrategien für w in p sprechen. (Allerdings kann es auf schlechte Strategien von s u.U. noch bessere von w geben. Um dies auszuschließen benötigt man den Begriff der global-optimalen Strategie und S4.5.) Es gilt dann nämlich

$$\Sigma_w^{[n]}(p) \neq \emptyset \quad \text{und} \quad \Sigma_w^{[n-1]}(p) = \emptyset \qquad \text{bzw.}$$

$$\sigma \in \Sigma_w^{[n]}(p) = \bigvee_{\sigma \in \Sigma_w(p)} \bigwedge_{\tau \in \Sigma_s} |\langle p;\sigma,\tau\rangle| \leq n \wedge \bigwedge_{\sigma \in \Sigma_w(p)} \bigvee_{\tau \in \Sigma_s} |\langle p;\sigma,\tau\rangle| \geq n .$$

Es besteht aber auch das Bedürfnis ein Verhalten für w im Fall 2 und 3 anzugeben, also eine optimale Verteidigungsstrategie, die den Verlust möglichst weit hinauszögert bzw. eine Remis-erhaltende Strategie. Dazu definieren wir in Anlehnung an die Anschauung:

D4.2 Wir beschränken uns auf w und erklären die Menge $\Sigma_w^*(p)$ (lokal-)optimaler Strategien für w in p durch:

$$\sigma \in \Sigma_w^*(p) \quad :\Longleftrightarrow \quad \begin{cases} \text{Fall 1:} & \sigma \in \Sigma_w^{[z(p)]}(p) \\ \text{Fall 2:} & |\langle p;\sigma,\tau\rangle| \doteq z(p) \quad \text{für alle } \tau \in \Sigma_s(p) \\ \text{Fall 3:} & \langle p;\sigma,\tau\rangle \notin \mathfrak{A}_s \quad \text{für alle } \tau \in \Sigma_s \end{cases} .$$

Unter der Menge der <u>global-optimalen Strategien</u> für w verstehen wir dann

$$\Sigma_w^* \; := \; \bigcap_{p \in P} \Sigma_w^*(p) \qquad .$$

Im Fall 1 ist die Existenz optimaler Strategien klar, nicht jedoch etwa im Fall 2, da sich aus

$$\bigvee_{\tau \in \Sigma_s(p)} \; \bigwedge_{\sigma \in \Sigma_w} |\langle p;\sigma,\tau\rangle| \leq n \;\wedge\; \bigwedge_{\tau \in \Sigma_s(p)} \; \bigvee_{\sigma \in \Sigma_w} |\langle p;\sigma,\tau\rangle| \doteq n$$

nicht unmittelbar eine gute Verteidigungsstrategie gegen alle $\tau \in \Sigma_s(p)$, d.h.

$$\bigvee_{\sigma \in \Sigma_w} \; \bigwedge_{\tau \in \Sigma_s(p)} |\langle p;\sigma,\tau\rangle| \doteq n$$

bestimmen läßt.
Wir wollen dies und darüber hinaus sogar $\Sigma_w^* \neq \emptyset$ beweisen. Im Fall 1 hat man dann ein Analogon zu S3.5. Da, wie sich durch einfache Gegenbeispiele zeigen läßt, S3.4 für optimale Gewinnstrategien falsch ist, funktioniert der Beweis von S3.5 aber nicht für den analogen Fall optimaler Strategie-Mengen.
Wir führen daher ein neues Hifsmittel ein:

<u>D4.3</u> Der <u>Teilgraph</u> von Γ <u>der optimalen Züge</u> für w

$$\zeta_w : P_w \to \mathcal{P}(P_s)$$

wird erklärt durch

$$\zeta_w(p) := \left\{ q \in \Gamma p \;\middle|\; \begin{array}{ll} z(q) = z(p)-1 \wedge \Sigma_w(q) \neq \emptyset & \text{im Fall 1} \\ z(q) = z(p) & \text{im Fall 2,3} \end{array} \right\}$$

und entsprechend $\zeta_s : P_s \to \mathcal{P}(P_w)$.

<u>HS4.1</u> Sei $p \in P_w$ mit $\Sigma_w(p) \neq \emptyset$ und $p' := \sigma(p)$. Dann gilt

$$\sigma \in \Sigma_w^*(p) \quad \Longrightarrow \quad \sigma \in \Sigma_w^*(p') \text{ mit } z(p') = z(p)-1$$

Beweis: Sei $n := z(p)$. Im Beweis von S3.8 wurde gezeigt

$$\sigma \in \Sigma_w^{[n]}(p) \quad \Longleftrightarrow \quad \sigma \in \Sigma_w^{[n-1]}(p')$$

Es bleibt also $z(p') = n-1$ nachzuweisen. Dazu nehmen wir an $\Sigma_w^{[n-2]}(p') \neq \emptyset$ d.h. $\bigwedge_{\tau \in \Sigma_s} |\langle p';\sigma',\tau\rangle| \leq n-2$ für ein $\sigma' \in \Sigma_w^{[n-2]}(p')$.
Dann ergibt sich für die Strategie $\sigma'' \in \Sigma_w$, erklärt durch

$$\sigma''(q) := \begin{cases} p' & \text{für } q = p \\ \sigma'(q) & \phantom{\text{für }} q \neq p \end{cases}$$

für jedes $\tau \in \Sigma_s$

$$|\langle p;\sigma'',\tau\rangle| = |(p,p') \circ \langle p';\sigma',\tau\rangle| \leq 1+(n-2) = n-1$$

und somit

$$\sigma'' \in \Sigma_w^{[n-2]}(p) \qquad \text{im Widerspruch zu } \Sigma_w^{[n-2]}(p) = \emptyset \; .$$

<u>S4.2</u> Eigenschaften der Zugzahlfunktion z:

(a) Sei $p \in P_w$.

Fall 1: $z(\sigma(p)) \begin{cases} \geq \\ = \end{cases} z(p)-1$ für $\begin{matrix} \sigma \in \Sigma_w(p) \\ \sigma \in \Sigma_w^{[z(p)]}(p) \end{matrix}$

Fall 2,3: $z(\sigma(p)) \leq z(p)$ für $\sigma \in \Sigma_w$

(b) Sei $p \in P_s$.

Fall 2: $z(\tau(p)) \begin{cases} \geq \\ = \end{cases} z(p)-1$ für $\begin{matrix} \tau \in \Sigma_s(p) \\ \tau \in \Sigma_s^{[z(p)]}(p) \end{matrix}$

Fall 1,3: $z(\tau(p)) \leq z(p)$ für $\tau \in \Sigma_s$

(c) $z(p) = 0 \Leftrightarrow p \in V_w^{(0)} \cup V_s^{(0)}$.

Beweis: (a) Fall 1 ergibt sich unmittelbar aus S3.8(w1) und HS4.1 und Fall 2 aus S3.8(s2). Fall 3 ist trivial, da $z(p) = \infty$.
(b) ist analog (a).
(c) $z(p) = 0 \Leftrightarrow \Sigma_w^{[0]}(p) \neq \emptyset \vee \Sigma_s^{[0]}(p) \neq \emptyset$ und mit HS3.7(2)
$\Leftrightarrow p \in V_w^{(0)} \cup V_s^{(0)}$.

<u>HS4.3</u> Sei $p \in P_s$. Dann gilt: $\bigcap_{p' \in \Gamma p} \Sigma_w^*(p') \subset \Sigma_w^*(p)$.

Beweis: Im Fall 1 ist der Beweis ähnlich dem von HS4.1. Wir zeigen nur den Fall 2.
Sei also $\Sigma_s(p) \neq \emptyset$ und $\sigma \in \bigcap_{p' \in \Gamma p} \Sigma_w^*(p') \neq \emptyset$. Dann gilt

$$|\langle p;\sigma,\tau\rangle| = |(p,\tau(p))\circ\langle\tau(p);\sigma,\tau\rangle| \geq 1+z(p') \quad \text{für alle } \tau\in\Sigma_s(p)$$

und mit S4.2(b) gilt

$$\geq z(p) \quad \text{für alle } \tau\in\Sigma_s(p).$$

Also hat man $\sigma\in\Sigma_w^*(p)$.

S4.4 Für den Teilgraphen der optimalen Züge gilt:

$$\mathfrak{z}_w(p) \neq \emptyset \quad \Leftrightarrow \quad p\notin V_w^{(0)} .$$

Beweis: Es bleibt zu zeigen "$\prec$". Sei also $p\in V_w^{(0)}$.

Fall 1: Ergibt sich aus HS4.1.

Fall 2: Sei $n := z(p)$. Mittels S3.8(s2) erhält man

$$\Sigma_s^{[n]}(p') \neq \emptyset \succ \Sigma_s^{[n]}(p') \neq \emptyset \quad \text{für alle } p'\in\Gamma p$$

$$\Sigma_s^{[n-1]}(p) = \emptyset \succ \Sigma_s^{[n-1]}(q) = \emptyset \quad \text{für ein } q\in\Gamma p$$

und somit $z(q) = z(p)$.

Fall 3: Mit S3.6(w1) erhält man

$$\Sigma_w(p) = \emptyset \succ \Sigma_w(q) = \emptyset \quad \text{für ein } q\in\Gamma p$$

und somit $z(q) = z(p)$.

Aus dem folgenden Satz ergibt sich nun speziell $\Sigma_w^* \neq \emptyset$.

S4.5 Charakterisierung von Σ_w^* :

$$\sigma^*\in\Sigma_w^* \quad = \quad \bigwedge_{p\in P_w\setminus V_w^{(0)}} \sigma^*(p)\in\mathfrak{z}_w(p)$$

und analog für $\tau^*\in\Sigma_s^*$.

Beweis: Wegen S4.4 ist diese Konstruktion global-optimaler Strategien erklärt. Die wesentliche Aussage ist, daß ein so bestimmtes, lokal-optimales σ^* in Σ_w^* liegt, d.h. global-optimal ist. Sei also $\sigma^*\in\Sigma_w$ mit $\sigma^*(p)\in\mathfrak{z}_w(p)$ für alle $p\in P_w\setminus V_w^{(0)}$.

Wegen HS4.3 können wir uns auf P_w beschränken und haben für jedes $p\in P_w$ zu zeigen $\sigma^*\in\Sigma_w^*(p)$.

Mit $p_k := (\tau\sigma^*)^k(p)\in P_w$ und $q_k := \sigma^*(p_k)\in P_s$ für $k\geq 0$

erhalten wir die Darstellung

$$\langle p;\sigma^*,\tau\rangle = (p_o,q_o,p_1,q_1,\dots)$$

auf die wir uns im Beweis beziehen. Weiter sei $n := z(p)$.

Fall 1: Sei $\tau\in\Sigma_s$ beliebig.

Durch Induktion erhält man

$$\Sigma_w(p_k) \neq \emptyset \qquad \text{für alle } k \geq o \ .$$

Wir weisen o.E. den ersten Schritt nach: Es ist $\Sigma_w(p_o) \neq \emptyset$ und nach Definition von $\mathfrak{z}_w$ dann auch $\Sigma_w(\sigma^*(p_o)) \neq \emptyset$. Mit S3.6(s2) ergibt sich nun $\Sigma_w(p_1) \neq \emptyset$.

Nun zeigen wir

(*) $$z(p_k) \leq n-k$$

durch Induktion.

$k=o$: $\quad z(p_o) = z(p) = n$

$k\to k+1$: Nach Konstruktion von σ^* ist

$$z(\sigma^*(p_k)) = z(p_k)-1$$

und nach S4.2(b), wegen $\Sigma_w(\sigma^*(p_k)) \neq \emptyset$,

$$z(\tau\sigma^*(p_k)) \leq z(p_k)-1$$

und somit

$$z(p_{k+1}) \leq z(p_k)-1 \leq (n-k)-1 = n-(k+1) \ .$$

Wegen (*) - das beinhaltet, daß w seine Strategie höchstens n-mal "wechseln" muß - und S4.2(c), wodurch die Folge $(p_o,q_o,..)$ mit einem $p\in V_s^{(o)}$ endet, gilt

$$\langle p;\sigma^*,\tau\rangle\in\mathfrak{A}_w \quad \text{und} \quad |\langle p;\sigma^*,\tau\rangle| \leq n$$

also $\quad \sigma^*\in\Sigma_w^*(p)$.

Fall 2: Sei $\tau\in\Sigma_s(p)$ beliebig.

Wir zeigen diesmal

(**) $$z(p_k) \geq n-k$$

$k=o$: $\quad z(p_o) = z(p) = n$

$k\to k+1$: Nach Konstruktion ist

$$z(\sigma^*(p_k)) = z(p_k)$$

und dann mit S4.2(b), da wegen S3.8(s2) gilt $\Sigma_w(\sigma^*(p_k)) \neq \emptyset$,

$$z(\tau\sigma^*(p_k)) \geq z(p_k)-1 \ .$$

Also hat man

$$z(p_{k+1}) \geq z(p_k)-1 \geq n-k-1 \ .$$

Mit (**) ergibt sich nun

$$|\langle p;\sigma^*,\tau\rangle| \geq n \qquad \text{d.h. } \sigma^*\in\Sigma_w^*(p) \ .$$

Fall 3: Sei $\tau \in \Sigma_s$ beliebig.
Wir betrachten wieder S4.2(b). Entweder gilt stets $z(\tau(p_k)) = z(p_k)$, dann ist stets $z(p_k) = \infty$ und somit

$$|\langle p;\sigma^*,\tau\rangle| = \infty ,$$

oder es gibt ein m mit $z(\tau(p_m)) < z(p_m)$, dann ist aber notwendig $\Sigma_w(\tau(p_m)) \neq \emptyset$ und (wegen HS3.2)

$$\langle p;\sigma^*,\tau\rangle \in \mathfrak{A}_w .$$

In jedem Fall gilt also $\langle p;\sigma^*,\tau\rangle \notin \mathfrak{A}_s$
d.h. $\sigma^* \in \Sigma^*_w(p)$.

Sei $\sigma^* \in \Sigma^*_w$ und $\tau^* \in \Sigma^*_s$ ein beliebiges Paar global-optimaler Strategien, so gilt gemäß D4.2 und S4.5 für die Zugzahl der Stellung p

$$z(p) = |\langle p;\sigma^*,\tau^*\rangle| .$$

Die Zugzahl der - bei optimalem Verhalten - längsten Partie ist also dem Gesamtspiel (P,Γ) zugeordnet und gibt einen Hinweis auf die Schwierigkeit des Spieles. Wir definieren

<u>D4.4</u> Sei $m := \max_{p \in P:\ z(p)<\infty} z(p)$, so sprechen wir von (P,Γ) als einem <u>m-zügigen Spiel</u>.

Zum Schluß erwähnen wir noch einige Beziehungen, die sich aus S4.5 ergeben.

<u>S4.6</u> Für die Zugzahlfunktion erhält man im

Fall1: $z(p) = \min_{\sigma \in \Sigma_w(p)} \max_{\tau \in \Sigma_s} |\langle p;\sigma,\tau\rangle| = \max_{\tau \in \Sigma_s} \min_{\sigma \in \Sigma_w(p)} |\langle p;\sigma,\tau\rangle|$

Fall2: $z(p) = \min_{\tau \in \Sigma_s(p)} \max_{\sigma \in \Sigma_w} |\langle p;\sigma,\tau\rangle| = \max_{\sigma \in \Sigma_w} \min_{\tau \in \Sigma_s(p)} |\langle p;\sigma,\tau\rangle|$.

Beim Beweis zeigen wir o.E. den Fall 1:
Wie nach D4.1 schon erwähnt, hat man mit $n := z(p)$

$$\Sigma_w^{[n]}(p) \neq \emptyset \quad \Leftrightarrow \quad \begin{cases} \text{(a)} & \bigvee\limits_{\sigma \in \Sigma_w(p)} \bigwedge\limits_{\tau \in \Sigma_s} \langle p;\sigma,\tau\rangle \leq n \\ \text{(b)} & \bigwedge\limits_{\sigma \in \Sigma_w(p)} \bigvee\limits_{\tau \in \Sigma_s} \langle p;\sigma,\tau\rangle \geq n \end{cases}$$

Aus (b) folgt für alle $\sigma \in \Sigma_w(p)$ $\qquad \max\limits_{\tau \in \Sigma_s} |\langle p;\sigma,\tau\rangle| \geq n$

und sodann mit (a) $\qquad \min\limits_{\sigma \in \Sigma_w(p)} \max\limits_{\tau \in \Sigma_s} |\langle p;\sigma,\tau\rangle| = n$.

Aus (a) folgt für alle $\tau \in \Sigma_s$ $\qquad \min\limits_{\sigma \in \Sigma_w(p)} |\langle p;\sigma,\tau\rangle| \leq n$.

Aber mit S4.5 gibt es ein τ^* mit $\qquad \min\limits_{\sigma \in \Sigma_w(p)} |\langle p;\sigma,\tau^*\rangle| = n$

und somit $\qquad \max\limits_{\tau \in \Sigma_s} \min\limits_{\sigma \in \Sigma_w(p)} |\langle p;\sigma,\tau\rangle| = n$.

Ein Analogon im Fall 3, etwa $z(p) = \inf\limits_{\sigma \in \Sigma_w} \sup\limits_{\tau \in \Sigma_s} |\langle p;\sigma,\tau\rangle|$ gilt aber nicht allgemein. Um die Beziehungen des S4.6 dennoch zu erweitern und um sie zu dualisieren, führen wir eine geeignete Bewertung von Stellungen und Partien ein, die zwischen Gewinn und Verlust nicht nur quantitativ, sondern auch qualitativ unterscheidet.

<u>D4.5</u> <u>Bewertung von Positionen</u> $\quad v : P \to \mathbb{Q}$:

$$v(p) := \begin{cases} \frac{1}{z(p)+1} & \\ \frac{-1}{z(p)+1} & \text{im Fall } 2 \\ 0 & \end{cases} \quad \begin{matrix} \text{Fall } 1 \\ \\ \text{Fall } 3 \end{matrix} .$$

Für festes $p \in P$ setzen wir

$v := v(p)$

und erklären (unter Verwendung von Σ_w, Σ_s als Indexmengen) eine Matrix $A = (a_{\sigma\tau})$ durch

$$a_{\sigma\tau} := \begin{cases} \frac{1}{|\langle p;\sigma,\tau\rangle|+1} & \in \mathfrak{A}_w \\ \frac{-1}{|\langle p;\sigma,\tau\rangle|+1} & \text{falls } \langle p;\sigma,\tau\rangle \in \mathfrak{A}_s \\ 0 & \notin \mathfrak{A}_w \cup \mathfrak{A}_s \end{cases} .$$

S4.7 Es gilt

$$v = \max_{\sigma \in \Sigma_w} \min_{\tau \in \Sigma_s} a_{\sigma\tau} = \min_{\tau \in \Sigma_s} \max_{\sigma \in \Sigma_w} a_{\sigma\tau} ,$$

d.h. das "Matrixspiel" A hat einen Sattelpunkt mit Wert v.
Für $\sigma^* \in \Sigma_w^*$, $\tau^* \in \Sigma_s^*$ ergibt sich $v = a_{\sigma^*\tau^*}$.

Beweis:Fall 1: Mit $n := z(p) = \frac{1}{v} - 1$ schreibt sich die Bedingung $|\langle p;\sigma,\tau\rangle| \leq n$ in der Form $a_{\sigma\tau} \geq v$ und wir erhalten

(a) $\bigvee_{\sigma} \bigwedge_{\tau} a_{\sigma\tau} \geq v$

(b) $\bigwedge_{\sigma} \bigvee_{\tau} a_{\sigma\tau} \leq v$.

Bei (b) besteht zunächst die Einschränkung $\sigma \in \Sigma_w(p)$. Für $\sigma \notin \Sigma_w(p)$ gilt aber trivialerweise (b), da dann $a_{\sigma\tau} \leq o$ und $v > o$ ist.
Entsprechend zum Beweis von S4.6 ergibt sich aus (b)

$$\min_{\tau} a_{\sigma\tau} \leq v \quad \text{für alle } \sigma$$

und zusammen mit (a)

$$\max_{\sigma} \min_{\tau} a_{\sigma\tau} = v .$$

Aus (a) und S4.5 ergibt sich

$$\max_{\sigma} a_{\sigma\tau} \geq v \quad \text{für alle } \tau \text{ ; } \max_{\sigma} a_{\sigma\tau^*} = v$$

und somit

$$\min_{\tau} \max_{\sigma} a_{\sigma\tau} = v .$$

Fall 2: Mit $n := z(p) = -\frac{1}{v} - 1$ erhält man

$$|\langle p;\sigma,\tau\rangle| \leq n \Leftrightarrow a_{\sigma\tau} \leq v$$

und somit

(a)' $\bigvee_{\tau} \bigwedge_{\sigma} a_{\sigma\tau} \leq v$

(b)' $\bigwedge_{\tau} \bigvee_{\sigma} a_{\sigma\tau} \geq v$.

Für $\tau \notin \Sigma_s(p)$ ist (b)' wieder erfüllt, da dann $a_{\sigma\tau} \geq o$ und $v < o$.
Analog Fall 1 ergibt sich $\max_{\sigma} a_{\sigma\tau} \geq v$ für alle τ und dann $\min_{\tau} \max_{\sigma} a_{\sigma\tau} = v$; sowie $\min_{\tau} a_{\sigma\tau} \leq v$ für alle σ mit $\min_{\tau} a_{\sigma^*\tau} = v$, also $\max_{\sigma} \min_{\tau} a_{\sigma\tau} = v$.
Fall 3, d.h. $v = o$:
Aus $\min_{\tau} a_{\sigma\tau} \leq o$ für alle σ und $\min_{\tau} a_{\sigma^*\tau} = a_{\sigma^*\tau^*} = o$ folgt $\max_{\sigma} \min_{\tau} a_{\sigma\tau} = o$
und aus $\max_{\sigma} a_{\sigma\tau} \geq o$ für alle τ, $\max_{\sigma} a_{\sigma\tau^*} = o$ folgt $\min_{\tau} \max_{\sigma} a_{\sigma\tau} = o$.

§5 Beziehung zwischen Gewinn- und Verluststellungen

Mit den Begriffen und Hilfsmitteln der §§ 3 und 4 läßt sich nun leicht für jede Position Gewinn (bzw. Verlust, gemäß der Verabredung "w gewinnt = s verliert") erklären. w gewinnt in $p \in P$ soll heißen, daß es in p eine Gewinnstrategie für w gibt. Unter Beachtung der Numerierung $P = \{p_1, \ldots, p_{N_w}, \ldots, p_N\}$, kann man Gewinn für alle Stellungen durch einen Booleschen Vektor über P beschreiben.

<u>D5.1</u> Wir erklären <u>Gewinnvektoren</u> für w bzw. s

$A, A^{[n]}, A^{(n)}$ und $B, B^{[n]}, B^{(n)} \in \mathcal{B}_{N,1}$:

1. Gewinn für w

$A[i] := (\Sigma_w(p_i) \neq \emptyset) = L$ heißt "w gewinnt in p_i"

$A^{[n]}[i] := (\Sigma_w^{[n]}(p_i) \neq \emptyset) = L$ heißt "w gewinnt in p_i in höchstens n Zügen"

$A^{(n)}[i] := (\Sigma_w^{[n]}(p_i) \neq \emptyset \wedge \Sigma_w^{[n-1]}(p_i) = \emptyset)$

$= L$ heißt "w gewinnt in p_i in (genau) n Zügen".

2. Gewinn für s

$B[i] := (\Sigma_s(p_i) \neq \emptyset)$ usw.- analog 1.

3. Remis

Falls in p_i weder w noch s gewinnt, d.h. $A[i] \vee B[i] = 0$, so heißt p_i eine Remis-Position.

Bemerkung: Nach der Bem. zu D3.4 schließen sich die Fälle 1. und 2. der Definition gegenseitig aus.
Die in §2 eingeführten o-Verluststellungen ordnen sich der jetzigen Definition unter, d.h. es gilt

$$p_i \in V_s^{(0)} \succ A[i] = L$$

$$p_i \in V_w^{(0)} \succ B[i] = L .$$

Vereinbarung: Gemäß der Zerlegung $P = \{p_1,\dots,p_{N_w}\} \dot{\cup} \{p_{N_w+1},\dots,p_N\}$ fassen wir die Mengen $V_w^{(o)}$ und $V_s^{(o)}$ hinfort in naheliegender Weise ($p_i \in V_w^{(o)} \text{ ☓ } V_w^{(o)}[i] = L$ und $p_i \in V_s^{(o)} \text{ ☓ } V_s^{(o)}[i-N_w] = L$) als Vektoren $V_w^{(o)} \in \mathcal{B}_{N_w,1}$ und $V_s^{(o)} \in \mathcal{B}_{N_s,1}$ auf.

Wir fassen die elementaren Eigenschaften der Gewinnvektoren in folgendem Hilfssatz zusammen, indem wir HS3.7 ergänzen.

<u>HS5.1</u> Eigenschaften der Gewinnvektoren (o.E. formuliert für A):

(1) $A^{[n-1]} \leq A^{[n]}$

(2) $A^{[o]} = A^{(o)} = \begin{pmatrix} 0 \\ \\ V_s^{(o)} \end{pmatrix}, \quad B^{[o]} = B^{(o)} = \begin{pmatrix} V_w^{(o)} \\ \\ 0 \end{pmatrix}$

(3) $\bigvee_{n \geq o} A^{[n]} = \bigvee_{n \geq o} A^{(n)} = A \qquad (\text{etwa } n \leq [\frac{N-1}{2}])$

(4) $A^{(n)} = A^{[n]} \wedge \bar{A}^{[n-1]}$ und $A^{[n]} = \bigvee_{k=o}^{n} A^{(k)}$

(5) $A^{(n)} \wedge A^{(m)} = 0 \qquad \text{für } n \neq m$.

Das hauptsächliche Problem besteht nun in der Bestimmung der Gewinnvektoren $A^{(n)}$ und $B^{(n)}$ $(n \geq o)$, die im wesentlichen die Information des Spieles (P,Γ) beschreiben. Eine global-optimale Strategie läßt sich bei einem praktischen Algorithmus leicht erhalten.

Vereinbarung: Die Zerlegung der Spielzustände $P = P_w \dot{\cup} P_s$ induziert eine Einteilung der Gewinnvektoren, die wir für A und B (und entsprechend dann für $A^{(n)}$ usw.) wie folgt schreiben wollen

$$A = \begin{pmatrix} G_w \\ V_s \end{pmatrix} \quad , \quad B = \begin{pmatrix} V_w \\ G_s \end{pmatrix} \quad .$$

Die Bezeichnung ist mnemotechnisch gewählt, so daß G bzw. V, Gewinn bzw. Verlust für (den Index) w oder s angeben, eine Partei, die dann am Zuge ist. Die Aussage "w gewinnt" bezieht sich also immer auf eine Position mit w am Zug usw.

Der Vorteil dieser Notation wird in folgendem Satz ersichtlich, der mit Hilfe der Sätze 3.6 und 3.8 Aufschluß über den Zusammenhang der Gewinnpunkte im Spielgraphen - dargestellt als Beziehung zwischen den Gewinnvektoren - gibt.

S5.2 Zusammenhang zwischen Gewinn- und Verluststellungen:

$$(1) \begin{cases} G_w = B_w \; V_s \\ V_s = \overline{B_s \; \overline{G}_w} \end{cases} \qquad (2) \begin{cases} G_s = B_s \; V_w \\ V_w = \overline{B_w \; \overline{G}_s} \end{cases} .$$

Beweis: Wir zeigen o.E.(1). Unter Verwendung von S3.6 erhält man für $p = p_i \in P_w$:

$$G_w[i] = A[i] = (\Sigma_w(p) \neq \emptyset) = (\bigcup_{p' \in \Gamma p} \Sigma_w(p') \neq \emptyset)$$
$$= \bigvee_j T[i,j] \wedge A[j] = \bigvee_j B_w[i,j] \wedge V_s[j]$$

und für $p = p_i \in P_s$:

$$V_s[i] = A[i] = (\Sigma_w(p) \neq \emptyset) = (\Sigma_w(p') \neq \emptyset \text{ für alle } p' \in \Gamma p)$$
$$= \bigwedge_j (T[i,j] \leqq A[j]) = \bigwedge_j (\overline{T}[i,j] \vee A[j])$$
$$= \overline{\bigvee_j T[i,j] \wedge \overline{A}[j]} = \overline{\bigvee_j B_s[i,j] \wedge \overline{G}_w[j]} .$$

Bei Benutzung des S3.8 - der S3.6 mit Beachtung der Zugzahlen entspricht - ergibt sich ganz analog, wobei wir nur die (1) entsprechenden Formeln angeben:

S5.3 Algorithmus zur Bestimmung der $A^{[n]}$:

(I) $\quad G_w^{[n]} = B_w \; V_s^{[n-1]}$

(II) $\quad V_s^{[n]} = \overline{B_s \; \overline{G}_w^{[n]}}$

mit Startvektor $V_s^{[o]} = V_s^{(o)}$.

<u>S5.4</u> Es besteht die Ungleichung

$$G_w^{(n)} \leq B_w\, V_s^{(n-1)} ,$$

bzw. genauer die Gleichung

$$G_w^{(n)} = B_w\, V_s^{(n-1)} \wedge \bar{G}_w^{[n-1]} ,$$

so daß sich also der Algorithmus (I) auch in der Form schreibt

$$\text{(I)'} \qquad G_w^{[n]} = G_w^{[n-1]} \vee B_w\, V_s^{(n-1)} .$$

Bemerkung: Ein entsprechendes Vorgehen bei (II) ist falsch, da i.a. nur gilt

$$V_s^{[n]} \geq V_s^{[n-1]} \vee \overline{B_s\, \bar{G}_w^{(n)}} .$$

Unter Beachtung von S1.4 erhält man nämlich

$$\begin{aligned} V_s^{[n]} &= \overline{B_s\, \bar{G}_w^{[n]}} = \overline{B_s\, \overline{(G_w^{[n-1]} \vee G_w^{(n)})}} = \overline{B_s\, (\bar{G}_w^{[n-1]} \wedge \bar{G}_w^{(n)})} \\ &\geq \overline{B_s\, \bar{G}_w^{[n-1]} \wedge B_s\, \bar{G}_w^{(n)}} = \overline{B_s\, \bar{G}_w^{[n-1]}} \vee \overline{B_s\, \bar{G}_w^{(n)}} \\ &= V_s^{[n-1]} \vee \overline{B_s\, \bar{G}_w^{(n)}} . \end{aligned}$$

Beweis von S5.4: Wir benutzen die Tatsache, daß die Booleschen Matrizen einen $\vee$-multiplikativen Verband bilden, und erhalten:

$$\begin{aligned} G_w^{(n)} &= G_w^{[n]} \wedge \bar{G}_w^{[n-1]} = B_w\, V_s^{[n-1]} \wedge \bar{G}_w^{[n-1]} \\ &= B_w\, (V_s^{[n-2]} \vee V_s^{(n-1)}) \wedge \bar{G}_w^{[n-1]} \\ &= (B_w\, V_s^{[n-2]} \vee B_w\, V_s^{(n-1)}) \wedge \bar{G}_w^{[n-1]} \\ &= (G_w^{[n-1]} \vee B_w\, V_s^{(n-1)}) \wedge \bar{G}_w^{[n-1]} \\ &= B_w\, V_s^{(n-1)} \wedge \bar{G}_w^{[n-1]} \\ G_w^{[n]} &= G_w^{[n-1]} \vee G_w^{(n)} = G_w^{[n-1]} \vee (B_w\, V_s^{(n-1)} \wedge \bar{G}_w^{[n-1]}) \\ &= G_w^{[n-1]} \vee B_w\, V_s^{(n-1)} . \end{aligned}$$

Eliminiert man im Algorithmus (I,II) V_s, so erhält man

S5.5 Rekursiver Aufbau der Folge $G_w^{[n]}$ beginnend mit $G_w^{[0]} = B_w V_s^{(0)}$:

$$G_w^{[n]} = B_w\, \overline{B_s\, \bar{G}_w^{[n-1]}} .$$

Setzt man dies über einen weiteren Schritt fort, so ergibt sich in

$$G_W^{[2]} = B_W \overline{B_S B_W B_S \bar{G}_W^{[0]}}$$

für festes $p \in P_W$ ein Verfahren diese Stellung auf Gewinn in zwei Zügen zu untersuchen.

S5.6 Ein Teilspiel eines Spieles erlaubt eine abgeschlossene Berechnung, d.h. die Formeln der Sätze 5.2 und 5.3 gelten für die entsprechenden Teilvektoren.

Beweis: Die Existenz eines Teilspieles (P',Γ') von (P,Γ) (mit $P' = P_W' \dot\cup P_S' \subset P$) besagt, daß sich die Matrizen B_W und B_S bezüglich der Zerlegungen

$$P_W = P_W' \dot\cup P_W'' \quad \text{und} \quad P_S = P_S' \dot\cup P_S''$$

in der Form

$$B_W = \left(\begin{array}{c|c} B_W' & 0 \\ \hline \multicolumn{2}{c}{B_W''} \end{array}\right) \quad \text{und} \quad B_S = \left(\begin{array}{c|c} B_S' & 0 \\ \hline \multicolumn{2}{c}{B_S''} \end{array}\right)$$

schreiben lassen. Zerlegt man G_W und G_S entsprechend

$$G_W = \begin{pmatrix} G_W' \\ G_W'' \end{pmatrix} \quad \text{und} \quad G_S = \begin{pmatrix} G_S' \\ G_S'' \end{pmatrix}$$

so erhält man in die Beziehung $G_W = B_W V_S$ eingesetzt

$$\begin{pmatrix} G_W' \\ G_W'' \end{pmatrix} = \begin{pmatrix} B_W' & 0 \\ & B_W'' \end{pmatrix} \cdot \begin{pmatrix} V_S' \\ V_S'' \end{pmatrix} = \begin{pmatrix} B_W' \cdot V_S' \\ B_W'' \cdot V_S \end{pmatrix}$$

Also insbesondere

$$G_W' = B_W' V_S' \quad .$$

Die Gültigkeit der übrigen Beziehungen für die Teilvektoren G_W', G_S', V_W', V_S' des Teilspiels läßt sich ebenso einfach nachprüfen.

§6 Die Kerne eines Spielgraphen

Bringen wir die Gleichungen des S5.2 auf die Form

$$(1)' \begin{cases} G_w = B_w V_s \\ \overline{V}_s = B_s \overline{G}_w \end{cases} \qquad (2)' \begin{cases} G_s = B_s V_w \\ \overline{V}_w = B_w \overline{G}_s \end{cases}$$

so legt dies die Einführung von Vektoren

$$X_1 := \begin{pmatrix} G_w \\ \overline{V}_s \end{pmatrix} \quad \text{und} \quad X_2 := \begin{pmatrix} \overline{V}_w \\ G_s \end{pmatrix}$$

nahe, wobei wir an die Definition

$$A = \begin{pmatrix} G_w \\ V_s \end{pmatrix} \quad , \quad B = \begin{pmatrix} V_w \\ G_s \end{pmatrix}$$

erinnern. S5.2 ist dann nämlich äquivalent mit

<u>S6.1</u> X_1 und X_2 sind Lösungen der Booleschen Gleichung

$$X = T\,\overline{X}$$

wobei $T = \begin{pmatrix} 0 & B_w \\ B_s & 0 \end{pmatrix}$ die assoziierte Matrix des Spielgraphen ist.

Bemerkung: Für eine beliebige Matrix T kann man feststellen: Ist die i-te Zeile von T eine Nullzeile, so ist $X[i] = 0$. Somit gilt speziell für die Lösungen der Booleschen Gleichung aus S6.1

$$X \leq \begin{pmatrix} \overline{V}_w^{(0)} \\ \overline{V}_s^{(0)} \end{pmatrix}$$

d.h. $\begin{pmatrix} V_w \\ V_s \end{pmatrix} \geq \begin{pmatrix} V_w^{(0)} \\ V_s^{(0)} \end{pmatrix}$ und $\begin{pmatrix} G_w \\ G_s \end{pmatrix} \leq \overline{\begin{pmatrix} V_w^{(0)} \\ V_s^{(0)} \end{pmatrix}}$

<u>S6.2</u> Die Lösungen X_1 und X_2 der Gleichung $X = T\,\overline{X}$ erhält man iterativ durch die Vorschrift

$$\text{(I)} \quad X_1^{[n]} = T\,\overline{T\,\bar{X}_1^{[n-1]}} \qquad \text{mit} \qquad X_1^{[0]} = \begin{pmatrix} 0 \\ \bar{V}_s^{(0)} \end{pmatrix}$$

$$\text{(II)} \quad X_2^{[n]} = T\,\overline{T\,\bar{X}_2^{[n-1]}} \qquad \text{mit} \qquad X_2^{[0]} = \begin{pmatrix} \bar{V}_w^{(0)} \\ 0 \end{pmatrix} .$$

Denn es ergibt sich

$$X_1^{[n]} = \begin{pmatrix} G_w^{[n]} \\ \bar{V}_s^{[n]} \end{pmatrix} \quad \text{und} \quad X_2^{[n]} = \begin{pmatrix} \bar{V}_w^{[n]} \\ G_s^{[n]} \end{pmatrix} ,$$

und somit löst man die Systeme des S5.3 und das Verfahren endet, falls $X_1^{[n+1]} = X_1^{[n]}$ bzw. $X_2^{[m+1]} = X_2^{[m]}$.

Beweis: Die doppelte Iteration $X = T\,\overline{T\,\bar{X}}$ von $X = T\,\bar{X}$ entspricht der Tatsache, daß einem Zug des Spieles ein Weg der Länge 2 im Spielgraphen zugeordnet ist. Der Beweis wird durch Iteration über n erbracht. Wir zeigen nur den Iterationsschritt für (I).

$$T\,\bar{X}_1^{[n-1]} = \begin{pmatrix} B_w & V_s^{[n-1]} \\ B_s & \bar{G}_w^{[n-1]} \end{pmatrix} = \begin{pmatrix} G_w^{[n]} \\ \bar{V}_s^{[n-1]} \end{pmatrix}$$

$$X_1^{[n]} = T\,\overline{T\,\bar{X}_1^{[n-1]}} = \begin{pmatrix} B_w & V_s^{[n-1]} \\ B_s & \bar{G}_w^{[n]} \end{pmatrix} = \begin{pmatrix} G_w^{[n]} \\ \bar{V}_s^{[n]} \end{pmatrix} .$$

Bemerkung: Man kann auch mit $X_1^{[0]} = \begin{pmatrix} 0 \\ \vec{L} \end{pmatrix}$ starten; dabei ergibt sich allerdings

$$T\,\bar{X}_1^{[0]} = \begin{pmatrix} 0 \\ B_s \cdot \vec{L} \end{pmatrix} = \begin{pmatrix} 0 \\ \bar{V}_s^{(0)} \end{pmatrix}$$

$$T\,\overline{T\,\bar{X}_1^{[0]}} = \begin{pmatrix} B_w & V_s^{(0)} \\ B_s & \vec{L} \end{pmatrix} = \begin{pmatrix} G_w^{[1]} \\ \bar{V}_s^{(0)} \end{pmatrix} = X_1^{[1]} .$$

Wir erinnern nun an den Begriff des Kerns aus §1. Ein Kern ist eine Menge, für die eine einschränkende Eigenschaft der Art "man kann hinein, aber man muß heraus" gilt. Diese Beziehung zwischen Können und Müssen drückt etwa die Verhältnisse bei Gewinn- und Verluststellungen aus. Der Zusammenhang ergibt sich unmittelbar aus S1.6.

S6.3 Der Spielgraph (P,Γ) besitzt zwei ausgezeichnete Kerne K_1 und K_2, nämlich

$$K_v = \{p_i \in P \mid X_v[i] = 0\} \qquad v = 1,2.$$

Korollar: Ausführlich geschrieben bedeuten diese Kerne

$K_1 = \{p \in P_w \mid \text{w gewinnt nicht}\} \cup \{p \in P_s \mid \text{s verliert}\}$

$K_2 = \{p \in P_w \mid \text{w verliert}\} \cup \{p \in P_s \mid \text{s gewinnt nicht}\}$

und somit ist (wegen $V_w \leq \bar{G}_w$, $V_s \leq \bar{G}_s$)

$K_1 \cap K_2 = \{p \in P_w \mid \text{w verliert}\} \cup \{p \in P_s \mid \text{s verliert}\}$

die Menge der Verluststellungen und

$K_1 \setminus K_2 = \{p \in P_w \mid \text{w hält Remis}\}$

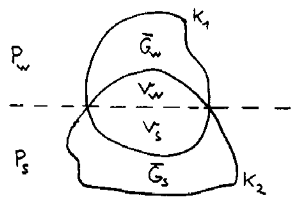
$K_2 \setminus K_1 = \{p \in P_s \mid \text{s hält Remis}\}$

die Menge der Spielremisen.

$K_1 \cup K_2 = \{p \in P_w \mid \text{w gewinnt nicht}\} \cup \{p \in P_s \mid \text{s gewinnt nicht}\}$

ist die Menge der Nicht-Gewinnstellungen.

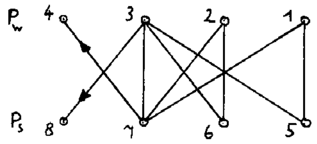
Wir können von der Lage der Kerne also folgende Skizze geben :



Dabei sind die Extremfälle $K_1 \cap K_2 = \emptyset$ (es gibt nur Remisstellungen) und $K_1 = K_2$ (es gibt keine Remisstellungen) möglich.

Ein zyklisch 2-geteilter Graph kann außer den Kernen K_1 und K_2 des S6.3 weitere Kerne haben.

Beispiel: Wir geben den Graphen durch sein Bild an.



Dabei bedeute ein Bogen (i,j) ohne Pfeil: $i \in \Gamma j$ und $j \in \Gamma i$

Für diesen Graphen ergeben sich die ausgezeichneten Kerne

$$K_1 = \{1,2,4,8\} \quad , \quad K_2 = \{4,5,6,8\} \, ,$$

sowie zwei weitere Kerne

$$K_3 = \{1,4,6,8\} \quad , \quad K_4 = \{2,4,5,8\} \, .$$

Jedoch sind die Kerne K_1 und K_2 charakteristisch in folgendem Sinne:

S6.4 Seien K_1 und K_2 die ausgezeichneten Kerne des Spielgraphen (P,Γ). Dann gilt für jeden beliebigen weiteren Kern K

$$K_1 \cap K_2 \subset K \subset K_1 \cup K_2 .$$

Bemerkung: Das bedeutet, daß durch einen beliebigen Kern (als einer allgemeinen Lösung des Booleschen Gleichungssystems $X = T\,\overline{X}$) u.U. die Remisstellungen nicht alle - im Gegensatz zu den Verluststellungen - beschrieben werden.

Beweis von S6.4: Jeder Kern enthält die Menge $\{p \in P \mid \Gamma p = \emptyset\}$ und somit gilt

(1) $\{p_i \in P_s \mid V_s^{(0)}[i] = L\} \subset K$.

Weiter bestehen die Implikationen

(2) Falls $\{p_i \in P_s \mid V_s^{[n-1]}[i] = L\} \subset K$ dann $\{p_i \in P_w \mid G_w^{[n]}[i] = L\} \subset \overline{K}$.

(3) Falls $\{p_i \in P_w \mid G_w^{[n]}[i] = L\} \subset \overline{K}$ dann $\{p_i \in P_s \mid V_s^{[n]}[i] = L\} \subset K$.

Wir zeigen zunächst (2). Sei $p = p_i$ mit $G_w^{[n]}[i] = L$. Dann gibt es nach S5.3(I) ein $p_j \in \Gamma p$ mit $V_s^{[n-1]}[j] = L$, d.h. $p_j \in K$ und somit $p \notin K$. Zum Nachweis von (3) sei $p = p_i$ mit $V_s^{[n]}[i] = L$ und wir nehmen an $p \notin K$. Das bedeutet $\Gamma p \cap K \neq \emptyset$ und somit gibt es ein $p_j \in \Gamma p \cap K$. Nach S5.3(II) ist aber $G_w^{[n]}[j] = L$, d.h. $p_j \notin K$. Dies ist ein Widerspruch.

Aus (1) ergibt sich mittels der Implikationen (2) und (3) durch Induktion und mit Hilfe der Beziehung $G_w = \bigvee_{k \geq 1} G_w^{[k]}$, $V_s = \bigvee_{k \geq 0} V_s^{[k]}$:

(4) $\{p_i \in P_s \mid V_s[i] = L\} \subset K$; $K \subset \{p_i \in P_w \mid G_w[i] = 0\}$

und entsprechend bei Betrachtung von V_w und G_s:

(5) $\{p_i \in P_w \mid V_w[i] = L\} \subset K$; $K \subset \{p_i \in P_s \mid G_s[i] = 0\}$.

Dies bedeutet aber unter Beachtung der im Anschluß an S6.3 erfolgten Erläuterungen

$$K_1 \cap K_2 \subset K \subset K_1 \cup K_2 .$$

§7 Vereinfachungen für ein spezielles Brettspiel

Die Vorstellung von den bekannten Brettspielen verbindet sich mit einer einfacheren Struktur als die des Spiels der allgemeinen Definition des §2. Wir wollen auf diese mögliche Vereinfachung in der Darstellung eingehen und orientieren uns dazu am Schachspiel.
Sehen wir von Besonderheiten - wie der Einschränkung des Zugrechtes einer Figur durch die Stellung anderer, sowie der Möglichkeit des Schlagens - ab, so können wir von folgender Situation ausgehen: Auf einem Schachbrett bewegen zwei Parteien einige Figuren. Jeder einzelnen Figur ist ein bestimmtes Zugrecht zugeordnet. Der Zug einer Partei erfährt nun folgende Einschränkungen (im Vergleich zu den möglichen Übergängen, die die Matrix T des Schachspiels erlauben könnte) :

(A) Eine Partei ändert nur die Stellung der eigenen Figuren,

(B) sie wählt genau eine Figur und bewegt sie gemäß deren Zugrecht.

Die Eigenschaft (B) gestattet die Beschränkung auf eine Figur (als "Summenfigur") für jede Partei durch Betrachtung des Summengraphen.

<u>Definition eines speziellen Brettspieles:</u>
Für zwei Parteien oder Figuren (1) und (2) sei je ein Graph

$G_1 = (Q_1,\Gamma_1)$, $G_2 = (Q_2,\Gamma_2)$ mit assoziierten Matrizen B_1,B_2

gegeben, der die Stellungen und Gangart der Figuren beschreiben soll, in dem Sinne "Figur (F) wird vermöge Γ_F auf Q_F gezogen".
Weiter sei eine beliebige "Zielmenge"

$$M \subset Q_1 \times Q_2 \quad \text{mit} \quad M = M_1 \cup M_2$$

zur Ergänzung der o-Verlust- oder Mattstellungen gegeben.
Ein Abspiel laufe folgendermaßen ab:
Beginnend mit einem Punkt $p_o=(i,k) \in Q_1 \times Q_2$ und einer Angabe darüber, ob (1) oder (2) in p_o am Zuge ist - damit ist p_o Ausgangsstellung des Spieles - werden die Figuren abwechselnd

gezogen, d.h.wenn etwa (1) in p_o am Zuge war, wählt (1) einen Punkt $i' \in \Gamma_1 i$ und sodann (2) einen Punkt $k' \in \Gamma_2 k$ usw. Das Spiel endet, falls eine der Figuren nicht mehr ziehen kann oder ein Punkt aus M erreicht ist und zwar mit Gewinn für (1), falls nach dem letzten Zug von (1) gilt: (2) kann nicht mehr ziehen, oder es ist ein Punkt aus M_2 erreicht. (Analog Gewinn für (2)).

Wir werden dieses Brettspiel nun in die Definition des §2 einordnen und die Ergebnisse des §5 in adäquater Form aufführen. Zunächst müssen wir dazu das Quadrupel (P_w, P_s, p_o, Γ) des Brettspieles mit w=(1) und s=(2) angeben.
Die Menge der Positionen besteht aus der disjunkten Vereinigung zweier Exemplare von $Q_1 \times Q_2$:

$$P_\nu := (Q_1 \times Q_2)_\nu \quad (\nu = 1,2) \ , \quad P := P_1 \dot{\cup} P_2 \quad ,$$

wobei der Index 1,2 angibt, welche Partei am Zug ist.
Vereinbarung: Diese Indizierung soll der Einfachheit halber nicht an der Position, sondern an den darauf operierenden "Funktionen" erfolgen.
Da die Positionen durch einen Produktraum beschrieben werden, verwenden wir in diesem Fall statt Gewinn- und Verlustvektoren, Matrizen, d.h. wir operieren mit $G_1, G_1^{(n)}, \dots, V_2^{[n]} \in \mathcal{L}_{|Q_1|, |Q_2|}$.
Wir schreiben etwa $V_1^{(o)}[i,k] = L$ mit der Bedeutung: $(i,k) \equiv (i,k)_1$ ist eine o-Verluststellung für (1) in der Stellung (i,k) mit (1) am Zuge.

Gemäß den Forderungen in der Definition des speziellen Brettspieles können wir weiter setzen

$$V_1^{(o)}[i,k] := (\Gamma_1 i = \emptyset \vee (i,k) \in M_1)$$

$$V_2^{(o)}[i,k] := (\Gamma_2 k = \emptyset \vee (i,k) \in M_2) \quad ,$$

sowie die Korrespondenz Γ auf P erklären vermöge

$$1. \ \Gamma | P_1 \qquad \Gamma(i,k) := \begin{cases} \Gamma_1 i \times \{k\} & \text{für } (i,k) \notin M_1 \\ \emptyset & \phantom{\text{für }} (i,k) \in M_1 \end{cases}$$

2. $\Gamma|P_2$ $\quad \Gamma(i,k) := \begin{cases} \{i\} \times \Gamma_2 k & \text{für} \quad (i,k) \notin M_2 \\ \emptyset & \qquad\;\; (i,k) \in M_2 \end{cases}$.

Diese Definitionen sind natürlich verträglich, da - wie man leicht sieht - gilt

$$V_\nu^{(o)}[i,k] = (\,\Gamma(i,k)_\nu = \emptyset\,) \qquad \text{für } \nu = 1,2.$$

Die Beschreibung des Zugrechtes eines Brettspieles erfolgt durch B_1 und B_2, die zusammen $2\cdot|Q_1|\cdot|Q_2|$ Elemente haben, während bei der allgemeinen Darstellung des §2 die Matrix T gleich $2\cdot(\,|Q_1|\cdot|Q_2|\,)^2$ wesentliche Elemente zählen würde.

Nun wollen wir die Sätze 5.3 und 5.4 anwenden und die für die hier eingeführten Matrizen geltenden Aussagen ableiten.

<u>S7.1</u> Beziehung zwischen den Gewinnmatrizen G_1 und V_2 eines speziellen Brettspieles

$$G_1^{(n)} = B_1\, V_2^{(n-1)} \wedge \bar{V}_1^{(o)} \wedge \bar{G}_1^{[n-1]}$$
$$V_2^{[n]} = \overline{\bar{G}_1^{[n]} B_2^{\,T}} \vee V_2^{(o)} \qquad .$$

<u>Korollar</u>: Deutet man (1) als Spinne und (2) als Fliege und $B_1 = B_2$ als Spinnennetz, so liefert S7.1 mit

$$G_1^{[o]} := V_1^{(o)} := E\,,\quad V_2^{(o)} := 0$$

einen Algorithmus zur Jagd "Spinne gegen Fliege".
Ein entsprechendes ALGOL-Programm von Prof.Dr.A.van Wijngaarden war Ausgangspunkt meiner Bemühungen kombinatorische Spiele zu analysieren.

Beweis von S7.1: Wir benutzen die Ungleichung $G_w^{(n)} \leq B_w\, V_s^{(n-1)}$ des S5.4 und erhalten aus $G_1^{(n)}[i,k] = L$ die Existenz eines p aus $\Gamma(i,k)_1$ mit $V_2^{(n-1)}[p] = L$. Dies ist aber gemäß Definition von $\Gamma|P_1$ sowie $V_1^{(o)}$ gleichbedeutend mit der Existenz eines $j \in \Gamma_1 i$ mit $V_2^{(n-1)}[j,k] = L$ und $(i,k)_1 \notin M_1$ bzw. $V_2^{(n-1)}[j,k] \wedge \bar{V}_1^{(o)}[i,k] = L$.
Wir haben also die Beziehung

$$\text{(I)} \qquad G_1^{(n)}[i,k] \leq \bigvee_{j \in \Gamma_1 i} V_2^{(n-1)}[j,k] \wedge \bar{V}_1^{(o)}[i,k]$$

gewonnen und damit auch die erste Gleichung bewiesen.
Entsprechend läßt sich aus $\bar{V}_s^{(n)} = B_s\ \bar{G}_w^{[n]}$ (S5.3) die zweite Gleichung ableiten:

$$\bar{V}_2^{T[n]}[k,i] = \bar{V}_2^{[n]}[i,k] = (\exists p \in \Gamma(i,k)_2 \mid \bar{G}_1^{[n]}[p] = L)$$
$$= \bigvee_{j \in \Gamma_2 k} \bar{G}_1^{[n]}[i,j] \wedge \bar{V}_2^{(o)}[i,k]$$
$$= \bigvee_j (B_2[k,j] \wedge \bar{G}_1^{T[n]}[j,i]) \wedge \bar{V}_2^{T(o)}[k,i]$$

Also $\bar{V}_2^{T[n]} = B_2\ \bar{G}_1^{T[n]} \wedge \bar{V}_2^{T(o)}$

$$\bar{V}_2^{[n]} = \bar{G}_1^{[n]} B_2^T \wedge \bar{V}_2^{(o)}$$
$$V_2^{[n]} = \overline{\bar{G}_1^{[n]} B_2^T} \vee V_2^{(o)} \quad \text{bzw.}$$
$$\text{(II)} \quad V_2^{[n]}[i,k] = \bigwedge_{j \in \Gamma_2 k} G_1^{[n]}[i,j] \vee V_2^{(o)}[i,k]$$

Wir betrachten die Aussagen des S7.1. Ist die Korrespondenz Γ_1 und somit die assoziierte Matrix B_1 abhängig von der Stellung k der Figur (2), so gilt nicht mehr die erste Gleichung, jedoch die Beziehung (I), die nur ein festes k enthält. Dazu muß man die Definition eines speziellen Brettspieles erweitern und Γ_1 durch $\Gamma_1(k)$ ersetzen, d.h. der Figur (1), $|Q_1|$ Graphen zuordnen. Entsprechendes gilt bei der Abhängigkeit von $\Gamma_2(i)$ für die Beziehung (II). (Diese Auffassung ist gegenüber D2.1 nur dann von Vorteil, wenn die $\Gamma_1(k)$ für alle k leicht zu beschreiben sind.)

Wir fassen dies in einem Satz zusammen, der Grundlage für einen praktischen Algorithmus ist.

<u>S7.2</u> Gegeben sei ein spezielles Brettspiel mit einer Abhängigkeit der Gangarten $\Gamma_1 = \Gamma_1(k)$ und $\Gamma_2 = \Gamma_2(i)$ von der Stellung der anderen Figur. Zur Berechnung der Gewinnmatrizen eignen sich die Beziehungen

$$\text{(I)} \begin{cases} G_1^{(n)}[i,k] = \bigvee_{j \in \Gamma_1 i} V_2^{(n-1)}[j,k] \wedge \bar{G}_1^{[n-1]}[i,k] \wedge \bar{V}_1^{(o)}[i,k] \\ G_1^{[n]} = G_1^{[n-1]} \vee G_1^{(n)} \end{cases}$$

$$(II)\ \begin{cases} V_2^{(n)}[i,k] = \bigwedge\limits_{j\in\Gamma_2 k} G_1^{[n]}[i,j] \wedge \overline{V}_2^{[n-1]}[i,k] \\ V_2^{[n]} = V_2^{[n-1]} \vee V_2^{(n)} \end{cases}$$

Zum Beweis beachte man bei (II) $V_2^{(o)} \leq V_2^{[n-1]}$ d.h. $V_2^{(o)} \wedge \overline{V}_2^{[n-1]} = 0$. Daraus ergibt sich dann gemäß S7.1 (II) und $V_2^{(n)} = V_2^{[n]} \wedge \overline{V}_2^{[n-1]}$ die Behauptung.

Bemerkung: Die Asymmetrie der Beziehungen, die sich im Vorko men von $\overline{V}_1^{(o)}$ in (I) ausdrückt, rührt von der Definition eines speziellen Brettspieles her, in der nur o-Verluststellungen (Matt) aber nicht o-Gewinnstellungen ausgezeichnet worden si

§8 Rückführung des Schachspieles auf ein spezielles Brettspiel

Wenden wir unsere Ergebnisse auf eines der bekanntesten Figuren-Brettspiele an: Das Schachspiel.
Beim Schach besteht eine Position $p = (w_1,..,w_{16},s_1,..,s_{16},F,I)$ aus den Lokalisierungen der Figuren $o \leq w_i,s_i \leq 64$ (= o bedeutet: Figur nicht vorhanden), der Angabe des Zugrechts F und einer geeigneten Information I zur Beachtung der Rochade, des En-Passant-Schlagens, der Zugwiederholungsregel und der 5o-Züge-Regel. Vom Parameter I wollen wir im weiteren absehen.
Geht man wie in §2 vor, so ist eine Beschreibung des Schachs nach entsprechender Berücksichtigung der Pattstellungen möglich. Wir wollen jedoch das Schachspiel als ein spezielles Brettspiel im Sinne des §7 auffassen und den Algorithmus des S7.2 auf seine Verwendbarkeit hin diskutieren.
Die Besonderheiten, die wir beachten müssen, sind die Definition des Matts, die Beschränkungen des Zugrechtes einer Figur durch andere, eingeschlossen der Möglichkeit des Schlagens, sowie der Regel, daß man nicht ins Schach ziehen darf, und das Patt. Wir streben an, den Schwierigkeiten durch geeignete Vorbesetzung der Startmatrix $V_s^{(o)}$ und konsequente Benutzung des Algorithmus Rechnung zu tragen.
Die Voraussetzung der Unabhängigkeit des Zugrechtes der Figuren voneinander ist nicht gegeben, und so lassen sich auch nicht mehrere Figuren zu einer Summenfigur zusammenfassen. Dennoch kann man alle Besonderheiten am speziellen (End-) Spiel

$$wK + wF \times sK + sF$$

studieren, bei dem wF bzw. sF gegebenenfalls als Repräsentant der weißen bzw. schwarzen Figuren steht.
Wir gehen also von folgender Situation aus:
Gegeben ist ein Schachbrett mit 64 Feldern von 1 bis 64 numeriert, sowie ein Feld o. Darauf operieren zwei Parteien, w und s mit je einer ausgezeichneten Figur, dem König K, und einer weiteren Figur F. Eine Stellung ist also gegeben durch ein

Quadrupel

(wK,wF,sK,sF) mit $1 \leq wK,sK \leq 64$, $o \leq wF,sF \leq 64$
sowie einer Angabe darüber, wer am Zuge ist.
Wir werden sehen, wie man auf das Feld o für geschlagene Figuren verzichten und das Schlagen einfacher berücksichtigen kann.

Weiter sei nun jeder einzelnen Figur ein bestimmtes Zugrecht zugeordnet. Zur Vereinfachung vereinbaren wir, die zugehörige Korrespondenz Γ nicht noch durch F zu indizieren, so daß also ΓF genau die Felder umfaßt, auf die die Figur F vom Feld F aus (unter Beachtung anderer Figuren) ziehen kann. Ebenso ist klar, was Γw bedeutet, wenn wir von der weißen Stellung w = (wK,wF) sprechen.
Das Zugrecht einer Figur erfährt durch die Stellung anderer Figuren genau definierte Veränderungen, zumeist Einschränkungen.
Zur Diskussion steht also der Algorithmus des S7.2

$$(\mathrm{I}) \begin{cases} G_w^{(n)}[w,s] = \bigvee_{wN \in \Gamma w} V_s^{(n-1)}[wN,s] \wedge \overline{G}_w^{[n-1]}[w,s] \wedge \overline{V}_w^{(o)}[w,s] \\ G_w^{[n]} = G_w^{[n-1]} \vee G_w^{(n)} \end{cases}$$

$$(\mathrm{II}) \begin{cases} V_s^{(n)}[w,s] = \bigwedge_{sN \in \Gamma s} G_w^{[n]}[w,sN] \wedge \overline{V}_s^{[n-1]}[w,s] \\ V_s^{[n]} = V_s^{[n-1]} \vee V_s^{(n)} \end{cases}$$

wobei Γ - sofern im folgenden nicht ausdrücklich anders vermerkt - das modifizierte Zugrecht bedeuten soll.

Ferner ist eine Angabe über die Startmatrizen $V_s^{(o)}$, $V_s^{[o]}$, $G_w^{[o]}$, $V_w^{(o)}$ nötig, die im wesentlichen auf die Vorbesetzung $V_s^{(o)} := M_s$ (Menge der Mattstellungen) hinausläuft.
Durch eine darüberhinausgehende Vorbesetzung wollen wir den Algorithmus jedoch so steuern, daß er einigen Besonderheiten des Schachs Rechnung trägt.

1. Einführung und Berücksichtigung illegaler Stellungen

Die Darstellung der Positionen als Elemente eines Produktraumes ermöglicht die Anlage bestimmter Stellungen, die beim Schachspiel illegal sind, nämlich solcher mit übereinstimmender Brettnummer (Inzidenzen), sowie mit benachbarten Königen.*)Im Hinblick auf eine maschinelle Realisierung ist es bequem, diese Stellungen - in denen es entsprechend der Betrachtung als spezielles Brettspiel ja auch ein Zugrecht gibt - hinzuzunehmen und dafür zu sorgen, daß kein Einfluß auf legale Stellungen entsteht. Dazu führen wir eine Matrix ILL ein, die genau diese Stellungen charakterisiert:

$$\begin{aligned} ILL\,[wK,wF,sK,sF] := {} & (wK = wF) \vee (wK = sK) \vee (wK = sF) \vee \\ & (wF = sK) \vee (wF = sF) \vee (sK = sF) \vee \\ & (sK \in \Gamma wK) \end{aligned}$$

S8.1 Die Vorbesetzung

(1) $$G_w^{[o]} := V_s^{[o]} := ILL$$

bewirkt im Algorithmus, daß illegale Stellungen nicht mehr als Gewinn- bzw. Verluststellungen erkannt werden,

d.h. $G_w^{(n)} \vee V_s^{(n)} \leq \overline{ILL}$ für $n \geq 1$.

Aus illegalen Stellungen darf also herausgezogen werden. Ebenfalls darf hineingezogen werden.

$V_s^{(1)}$ beschreibt die Menge der Stellungen $\{(w,s)_s \mid \Gamma s = \emptyset\}$, in denen s also nicht ziehen kann.

Beweis: Aus (I) ergibt sich (die bekannte Beziehung)

$$G_w^{(n)} \leq \overline{G}_w^{[n-1]}$$

und zusammen mit

$$ILL = G_w^{[o]} \leq G_w^{[n]} \quad \text{bzw.} \quad \overline{G}_w^{[n]} \leq \overline{ILL}$$

folgt also $G_w^{(n)} \leq \overline{ILL}$. Analog ergibt sich $V_s^{(n)} \leq \overline{ILL}$.

Wir haben uns nur noch zu überzeugen, daß auch durch Züge in illegale Stellungen keine Nebeneffekte auftreten. Entsprechend dem asymmetrischen Aufbau des Algorithmus bleibt nur die Wirkung von $G_w^{[o]}$ auf (II) zu untersuchen. In der Zerlegung

$$V_s^{(n)}[w,s] = \bigwedge_{\substack{sN \in \Gamma s \\ ILL[w,sN] = O}} G_w^{[n]}[w,sN] \wedge \bigwedge_{\substack{sN \in \Gamma s \\ ILL[w,sN] = L}} G_w^{[n]}[w,sN] \wedge \overline{V}_s^{[n-1]}[w,s]$$

*) Positionen, bei denen die ziehende Partei im Schach steht, gelten bei unserem Aufbau nicht als illegale. (S.Abschnitt 3.)

ist der zweite Faktor - wegen $G_w^{[n]} \geq G_w^{[o]}$ - identisch L und somit ohne Einfluß auf den Wert von $V_s^{(n)}[w,s]$, falls $\Gamma s \neq \emptyset$. Ist aber $\Gamma s = \emptyset$ für $(w,s)_S$ (und $V_s^{(o)}[w,s] = 0$), so ergibt sich bei der ersten Iteration $V_s^{(1)}[w,s] = L$.

Die Matrizen der Stufe o sind also im wesentlichen nur zum Steuern und nicht im Sinne der D2.1 zu verstehen. So ist etwa $V_s^{(o)} \neq V_s^{[o]}$ und $V_s^{(1)}$ entspricht $V_s^{(o)}$ usw.

Auf $V_s^{(1)}$ stehen durch die Vorbesetzung (1) schon die Stellungen mit $\Gamma s = \emptyset$, und es bleibt noch unsere Aufgabe durch $V_s^{(1)}$ auch die eigentlichen Mattstellungen beschreiben zu lassen. Wir werden Matt berechnen und streben dabei an, auf Inzidenzen (die überschaubar sind) zurückzugehen und den Algorithmus zu verwenden.

2. Rückführung des Matts auf Inzidenzen und Beachtung der Patt-Stellungen

Definition: Man sagt "sK steht im Schach", falls den König eine weiße Figur angreift, d.h.

$$sK \in \Gamma wF$$

(wobei Γ auch als Funktion von wK und sF zu verstehen ist). Eine Stellung, in der s am Zug ist und nicht ziehen kann oder sich durch jeden Zug ins Schach stellen würde, heißt für s "Matt", falls sK im Schach steht und ansonsten "Patt". Wenn dies durch die Booleschen Matrizen M_s bzw. $Patt_s$ charakterisiert wird, und wir für eine Stellung (w,s) die Aussagen

$$A := (sK \in \Gamma wF)$$
$$B := (\Gamma s = \emptyset) \vee (sKN \in \Gamma wF \text{ für alle } sN \in \Gamma s)$$

formulieren, so können wir schreiben

$$M_s[w,s] := A \wedge B$$
$$Patt_s[w,s] := \bar{A} \wedge B \ .$$

(Dabei ist Γ wieder so zu verstehen, daß gegebenenfalls gilt $\Gamma(w,s)_S = \{(wK,wF,sKN,sF), (wK,wF,sK,sFN), (wK,o,sK,sFN)\}$).

Die Berechnung von Matt und Patt aus den Inzidenzen erreichen wir nun durch Einführung des König-Schlagezuges "wF schlägt sK" d.h. durch die Vorbesetzung

(2) $\qquad V_s^{(0)}[w,s] \quad := \quad (wF = sK)$.

S8.2 Nach Vorbesetzung (1) und (2) und Anwendung des Algorithmus erhält man

$$G_w^{(1)}[w,s] \quad = \quad (sK \in \Gamma wF)$$
$$V_s^{(1)} \quad = \quad M_s \vee Patt_s \quad ,$$

so daß sich also für Matt und Patt ergibt

$$M_s \quad = \quad V_s^{(1)} \wedge G_w^{(1)}$$
$$Patt_s \quad = \quad V_s^{(1)} \wedge \bar{G}_w^{(1)}$$

Beweis: Vernachlässigen wir $V_w^{(0)}$ (siehe S8.5) und betrachten (I), so gilt $G_w^{(1)}[wK,wF,sK,sF] = L$ genau dann, wenn Bedingung ((a) oder (b)) und (c) erfüllt ist. Dabei ist

(a) $\exists\ wKN \in \Gamma w$ mit $V_s^{(0)}[wKN,wF,sK,sF] = L$ d.h. $wF = sK$

(b) $\exists\ wFN \in \Gamma w$ mit $wFN = sK$

(c) $G_w^{[0]}[wK,wF,sK,sF] = 0$.

Da (a) und (c) wegen (1) unvereinbar sind und (b) mit $sK \in \Gamma wF$ äquivalent ist, charakterisiert $G_w^{(1)}$ also die Stellungen, in denen sK im Schach steht.

Betrachten wir nur legale Stellungen, d.h. $V_s^{[0]}[w,s] = 0$, so gilt

$$V_s^{(1)}[w,s] = \bigwedge_{sN \in \Gamma s} G_w^{[1]}[w,sN] = (sKN \in \Gamma wF \text{ für alle } sN \in \Gamma s) .$$

Zusammen mit Vorbesetzung (1) und S8.1 ergibt dies die Bedingung B in der Definition von Matt und Patt und somit

$$V_s^{(1)} = M_s \vee Patt_s .$$

Durch (2) erscheinen also die Pattstellungen bei den 1-Verluststellungen. Dies muß richtiggestellt werden, was etwa nach der ersten Iteration durch den Prozeß

$$V_s^{(1)} := V_s^{(1)} \wedge G_w^{(1)}$$

(aber nachdem $V_s^{[1]} := V_s^{[0]} \vee V_s^{(1)}$ durchgeführt wurde!)

erreicht wird. Wir wollen diesen Sonderalgorithmus nicht einschieben und treffen unter der Annahme, daß $Patt_s$ von Anfang an bekannt ist, die Vereinbarung

(3) $$Patt_s \leftarrow V_s^{(0)} \quad .$$

Durch diese Vorbesetzung wird entsprechend der Berücksichtigung der illegalen Stellungen ohne Nebeneffekte erreicht, daß Pattstellungen nicht mehr betrachtet werden.

3. Vermeidung des Schlagens durch geeignete Einführung der Teilspiele

Betrachten wir die Menge der Stellungen

$$\{(wK,wF,sK,o) \mid wK,wF,sK \text{ beliebig}\} ,$$

so ist das dadurch gegebene Spiel $wK + wF \times sK$ ein Teilspiel zu $wK + wF \times sK + sF$ im Sinne der D2.2. Auf die Verlustmatrizen dieses Teilspiels beziehen wir uns mit $v_s^{(n)}$.
Die Gewinnmatrix des Teilspiels $wK \times sK + sF$ bezeichnen wir mit g_w.
Nach S5.6 können diese Teilspiele vorweg behandelt werden.
Wir fragen zunächst, was "Schlagen" bedeutet. Erstens die Einführung eines Feldes o für die geschlagene Figur und zweitens bei jedem Zug (in der innersten Schleife) eine Verzweigung der Art

$$\lceil wF \rceil (wK,wFN,sK,sF) = \begin{cases} (wK,wFN,sK,o) & \text{falls } wFN = sF \\ \text{ungeändert} & \text{sonst} \end{cases} \quad .$$

Und dies bei jeder Iteration, obwohl im wesentlichen das Teilspiel $wK + wF \times sK$ nur einmal betrachtet werden müßte. Wir werden uns im folgenden bemühen, die Information des Teilspiels so zu übernehmen, daß man auf das Schlagen verzichten kann.

a) Beachtung des Zuges "w schlägt sF":

Da ein weißer Zug zu untersuchen ist, betrachten wir Algorithmus (I) und fragen, wann ein Schlagezug von Bedeutung ist.
Dies drückt sich gemäß (I) genau in folgendem aus, wobei wir etwa "wK schlägt sF" betrachten:

(*) $V_s^{(n)}[wKN,wF,sK,o] = L \succ G_w^{(n+1)}[wK,wF,sK,wKN] = L$

für alle $wK \in \Gamma^{-1}wKN$

unter der Voraussetzung $G_w^{[n]} \vee V_w^{(o)}[wK,wF,sK,wKN] = 0$.

Das gleiche leistet aber der Algorithmus (I) und zwar ohne Schlagen, wenn wir dem aus (II) sich ergebendem $V_s^{(n)}$, die (*) entsprechenden Inzidenzsstellungen $(wKN,wF,sK,wKN)_s$ hinzufügen.

Wir bilden also

(4) $$V_s^{(n)} := V_s^{(n)} \vee V_s^{*(n)} ,$$

wobei V_s^* die Information über Verlust aus dem Teilspiel $wK + wF \times sK$ einführt

(5) $V_s^{*(n)}[wK,wF,sK,sF] = (wK=sF \vee wF=sF) \wedge V_s^{(n)}[wK,wF,sK]$ $(n \geq 1)$.

Vereinbarung: Da man bei den von uns gerechneten Schachendspielen nicht so sehr an der exakten Zugzahl bis zum Matt wie an der Zugzahl bis zu einem Teilspiel interessiert ist, haben wir durch folgende Zusammenfassung den Algorithmus vereinfacht. (Im günstigsten Fall verringert sich die Zahl der Iterationen dabei um die maximale Zugzahl der Teilspiele und man spart die ständige Durchführung des Prozesses (11).)

(4)' $$V_s^* \leq V_s^{(o)} \quad \text{mit} \quad V_s^* := \bigvee_{k \geq 1} V_s^{*(k)} .$$

Wir fassen zusammen

<u>S8.3</u> Durch (4)' wird die Information aus dem Teilspiel $wK + wF \times sK$ übernommen und die Schlagezüge "w schlägt sF" sind entbehrlich.

b) Beachtung der Züge "s schlägt wF" und "sF schlägt wK":

Der Beachtung dieser Schlagezüge liegt eine andere Auffassung zugrunde als die der weißen Schlagezüge unter a). Wir betrachten o.E. "sF schlägt wF" und stellen fest, daß allgemein gilt - und darin drückt sich die Bedeutung des Schlagezuges aus -

(**) $G_w[wK,\emptyset,sK,sFN]=0 \succ V_s[wK,sFN,sK,sF]=0$ für alle $sF \in \Gamma^{-1}sFN$

D.h. in der Stellung $(wK,sFN,sK,sF)_s$ verliert s nicht, da es zumindest einen Schlagezug hat, der in eine Stellung aus dem Teilspiel $wK \times sK + sF$ führt, die für w nicht gewonnen ist. Um dieses Schlagen gemäß (II) zu verhindern (nur um immer wieder festzustellen, daß s in diesen Stellungen einen guten Schlagezug hat), werden wir diese Stellung als o-zügige Verluststellung (!) erklären und dadurch analog der Beachtung des Patts und der illegalen Stellungen erreichen, daß sie nicht mehr behandelt wird.
Da wir (II) nicht in der (I) entsprechenden Form ($\bar{V}_s = \bigvee \bar{G}_w$) haben, müssen wir dazu einen eigenen Algorithmus formulieren

$$\text{(III)} \qquad D[w,s] \;:=\; \bigvee_{sN \in \Gamma s} C[w,sN] \quad .$$

Aus den (**) entsprechenden Inzidenzstellungen $(wK,sFN,sK,sFN)_w$ bzw.

$$\text{(6)} \quad C[wK,wF,sK,sF] := (sK{=}wF \vee sF{=}wF) \wedge \bar{g}_w[wK,sK,sF] \vee (sF{=}wK)$$

erhält man dann mittels (III) die in Frage stehenden Stellungen auf D. Läßt man illegale Stellungen außer Betracht, gilt nämlich für eine Stellung $(wK,wF,sK,sF)_s$, $D = L$ genau dann, wenn
$\exists\; sKN \in \Gamma sK$ mit $sKN{=}wF$ und $G_w[wK,o,sKN,sF] = 0$
oder $\exists\; sFN \in \Gamma sF$ mit $sFN{=}wF$ und $G_w[wK,o,sK,sFN] = 0$
oder $\exists\; sFN \in \Gamma sF$ mit $sFN{=}wK$.

<u>S8.4</u> Durch (6) und Algorithmus (III) ergibt sich die Matrix D, die die Information des Teilspiels $wK \times sK + sF$ einführt. Genügt die Vorbesetzung der Bedingung

$$\text{(7)} \qquad D \leq V_s^{[o]} \quad ,$$

so werden die schwarzen Schlagezüge "s schlägt wF" und "sF schlägt wK" entbehrlich.

Im Algorithmus (I) hat $V_w^{(o)}$ berücksichtigt werden müssen. Durch $V_w^{(o)} \leq G_w^{[o]}$ ließe sich (I) zu einer (II) entsprechenden Form vereinfachen, würde jedoch die Berechnung von $V_w^{(o)}$ voraussetzen. Nach der Vorbesetzung (7) erhält man auch so

S8.5 Berücksichtigung der weißen Mattstellungen:
Es genügt Algorithmus (I) in der Form

$$(I)' \quad G_w^{(n)}[w,s] = \bigvee_{wN \in \Gamma w} V_s^{(n-1)}[wN,s] \wedge \overline{G}_w^{[n-1]}[w,s]$$

durchzuführen.

Beweis: Ist $(w,s)_w$ eine weiße Mattstellung, so heißt das, daß $\Gamma w = \emptyset$ ist - und in diesem Fall ist (I)' äquivalent (I) - oder daß für alle $wN \in \Gamma w$ gilt $wKN \in \Gamma sF$ und somit $D[wN,s] = L$. Wegen (7) folgt weiter $V_s^{(n)}[wN,s] = 0$ für alle $n \geq 1$ und somit kann auch bei Algorithmus (I)' keine Mattstellung als Gewinnstellung erkannt werden, d.h.

$$G_w^{(n)} \leq \overline{V}_w^{(0)} \qquad \text{auch bei (I)'}.$$

4. Der Begriff der Rückrechnung

Betrachtet man den Algorithmus (I',II) für die Gesamtheit der Nicht-Remis-Stellungen, so erweist sich der Versuch, von bekannten Stellungen weiter- oder rückzurechnen, als vorteilhaft.

a) für Weiß

Algorithmus (I) oder (I)' unterscheidet sich von (II) in folgendem: Bei (I)' ist es nicht nötig, alle Stellungen (w,s) zu untersuchen, sondern es genügt von den Stellungen (wN,s) mit $V_s^{(n-1)}[wN,s] = L$ auszugehen und hat gemäß der Feststellung

$$(\alpha) \quad V_s^{(n-1)}[wN,s] \wedge \overline{G}_w^{[n-1]}[w,s] = G_w^{(n)}[w,s] \qquad \text{für alle } w \in \Gamma^{-1} wN.$$

alle sich ergebenden n-zügigen Gewinnstellungen.
Diese für einen praktischen Algorithmus wichtige Eigenschaft wollen wir als Rückrechnung für Weiß bezeichnen ("$G_w^{(n)}$ wird aus $V_s^{(n-1)}$ rückgerechnet").

b) für Schwarz

Das gleiche Vorgehen ist auch hier möglich, wenn man mit den komplementären Matrizen arbeitet und dafür den Algorithmus (II) formuliert

$$(II)' \quad \overline{V}_s^{(n)}[w,s] = \bigvee_{sN \in \Gamma s} \overline{G}_w^{[n]}[w,s] \vee \overline{V}_s^{[n-1]}[w,s] \quad .$$

Im Sinne von a) würde das bedeuten "$\overline{V}_s^{(n)}$ wird aus $\overline{G}_w^{[n]}$ rückgerechnet". Will man jedoch nicht zu den komplementären Matrizen übergehen, kann man die Beziehung

$$(\beta) \quad V_s^{(n)}[w,s] \leq \bigvee_{sN \in \Gamma s} G_w^{(n)}[w,sN]$$

ausnutzen. (Ihr liegt die Überlegung zu Grunde, daß eine Stellung (w,s) nur dann als n-zügige Verluststellung in Betracht kommt und mittels (II) untersucht werden muß, wenn mindestens für ein $sN \in \Gamma s$, (w,sN) als n-zügige Gewinnstellung erkannt worden ist.) Aus der Identität

$$V_s^{(n)}[w,s] = \bigwedge_{sN \in \Gamma s} G_w^{[n]}[w,sN] \wedge \overline{V}_s^{[n-1]}[w,s]$$
$$= \bigwedge_{sN \in \Gamma s} G_w^{[n]}[w,sN] \wedge \bigvee_{sN \in \Gamma s} G_w^{[n-1]}[w,sN]$$

und der Annahme $V_s^{(n)}[w,s] = L$ folgt sofort, daß es ein $sN \in \Gamma s$ gibt, mit $\overline{G}_w^{[n-1]} \wedge G_w^{[n]}[w,sN] = L$ d.h. $G_w^{(n)}[w,sN] = L$ und (β) ist bestätigt.

Wir können also ausgehend von $G_w^{(n)}$ mit $G_w^{(n)}[w,sN] = L$ die Stellungen (w,s) für alle $s \in \Gamma^{-1} sN$ markieren (z.B. auf $V_s^{(n)}$) und dann mittels (II) nur markierte Stellungen untersuchen. Dies wollen wir als Rückrechnung für Schwarz aus $G_w^{(n)}$ bezeichnen.

Bemerkung: Die "schwarze Rückrechnung" ist natürlich nur dann sinnvoll, wenn nur ein kleiner Teil der Gesamtstellungen markiert wird. In günstigen Fällen kann dadurch der Rechenaufwand auf einen Bruchteil zurückgehen.

5. Beschränkungen des Zugrechtes

Die bisherigen Überlegungen haben gezeigt, daß die Beschränkungen des Zugrechtes zum Teil entfallen können. Gemäß S8.1 ergibt sich

(8) König und Springer müssen keine Figur beachten, d.h. das zugehörige Γ erfährt keinerlei Beschränkungen.

Bemerkung: Bei der Feststellung in S8.1, daß $V_s^{(1)}$ die Stellungen mit $\Gamma s = \emptyset$ beschreibt, wird allerdings benützt, daß Γ das beschränkte Zugrecht darstellt. Im Falle eines Vier-Figuren-spieles gilt aber $\{(w,s)_s \mid \Gamma s = \emptyset\} = \emptyset$.

(9) Auf einen König fremder Farbe muß nicht achtgegeben werden.

Durch Vorbesetzung (2) und (6) wurden diese Stellungen nämlich schon behandelt.
Bei der folgenden Eigenschaft nutzen wir wesentlich aus, daß es sich um ein Vier-Figurenspiel handelt.

(1o) In einem 4-Figurenspiel wK + wF x sK + sF , bei dem sF eine solche Figur ist, daß die Veränderung des Zugrechts durch andere Figuren in einer Beschränkung des Zugrechts liegt, gilt darüberhinaus : sF muß nicht auf wF achten.

Beweis: sF beachtet wF heißt also: Falls $wF \in \Gamma sF$ erfährt Γ gewisse Beschränkungen zu $\Gamma(wF)$, d.h. $\Gamma(wF)sF \subset \Gamma sF$.
(Für sF = sB ist dies z.B. nicht richtig!)
Für ein 4-Figurenspiel gilt nun trivialerweise, daß für alle Stellungen ohne wF, w nicht gewonnen hat, d.h.

$$g_w[wK,sK,sF] \equiv 0$$

und somit nach S8.4

$$D[wK,wF,sK,sF] = L \qquad \text{falls} \quad wF \in \Gamma sF \ .$$

Das bedeutet aber wegen Vorbesetzung (7) $V_s^{[o]}[wK,wF,sK,sF] = L$ für alle Stellungen, in denen sF auf wF achten müßte.

Bemerkung: Umgekehrt gilt aber "wF beachtet sF" - wie man sich an trivialen Beispielen überlegt. Die obige Argumentation trifft deshalb nicht zu, weil $v_s[wK,wF,sK]$ nicht identisch L ist.

§9 Realisierung des Algorithmus für ein Schachendspiel auf einer Rechenanlage

Die Berücksichtigung der Besonderheiten des Schachs im vorigen Paragraphen, die im Hinblick auf eine schnelle maschinelle Realisierung vorgenommen worden ist, setzt in der beschriebenen Art voraus, daß der gesamte Stellungsraum betrachtet wird. Der gewonnene Algorithmus kann als Rückrechnung, ausgehend von Mattstellungen bezeichnet werden, während die für beliebiges n bei fester Stellung ausgeführte Formel des S5.5 als Vorwärtsrechnung (Aufstellen des Spielbaumes) angesehen werden kann. Eine Rückrechnung ohne vorherige Anlage aller Stellungen ist im Falle weniger Nicht-Remisstellungen auch durch Auflisten möglich.

Auf die Bedeutung und die Grenzen des Verfahrens in §8 wollen wir kurz eingehen. Wir akzeptieren die Fragestellung nach der exakten Zugzahl. Man kann ein triviales Spiel angeben, bei dem eine Strategie für alle Stellungen erklärt sein muß (, da alle bei der Partie durchlaufen werden).

Betrachten wir ein nicht triviales Beispiel: das Turm-Endspiel wK + wT $\times$ sK . Bei der Methode der Rückrechnung hat man alle Stellungen, das sind 2^{18}, zu betrachten. Weiter gilt $|\Gamma w| \leq 22$ und $|\Gamma s| \leq 8$, und wir nehmen als Durchschnittswerte 18 bzw. 7 an. Bei der Vorwärtsrechnung entspricht nun "$B_w \cdot B_s$" einem Zug und der Anlage von $7 \cdot 18 = 126 \approx 2^7$ Stellungen im Spielbaum.

Wir fragen nun, ausgehend von einer festen Stellung: Wieviel Züge (x) exakter Vorwärtsrechnung sind nötig, um ebensoviele Stellungen anzulegen, wie bei der Rückrechnung zu betrachten sind, nämlich 2^{18}?

Aus der Beziehung $(2^7)^x = 2^{18}$ ergibt sich $x = \frac{18}{7} < 3$. Also hat man bei 3 Zügen optimaler Vorwärtsrechnung schon mehr Stellungen anzulegen als die Gesamtheit aller Stellungen ausmacht.

Der Algorithmus (I,II) in der Form des §8 wird sich also besonders für Probleme eignen, bei denen exakte Zugzahl und eventuell eine Gesamt-Ergebnisliste interessieren.

Ein anderer Weg muß beim Schach beschritten werden, falls ein Programm für beliebig viele Figuren gesucht wird. Unter Verzicht auf Optimalität wird dabei die Vielzahl der zu untersuchenden Stellungen des Spielbaumes in mehrfacher Weise eingeschränkt, im wesentlichen nach einer Methode, die man wie folgt beschreiben kann: Selektive (bewertete) Vorwärtsrechnung (eventuell wird von bestimmten "toten" Stellungen nicht mehr weiter gerechnet) in einer variablen Tiefe (3 - 6 Züge, je nach Stellung) ; anschließend Bewertung der erhaltenen Stellungen durch eine empirisch eingegebene Funktion.
Für eine Übersicht über diese Bemühungen um ein allgemeines Schachprogramm verweisen wir auf folgende Literatur:

Samuel A.L.: Programming Computers to Play Games, in Advances in Computers, by Alt, Acad.Press, 196o, S.165-192.

Michie D.: Game-playing and Game-learning Automata, in Advances in Programming and Non-numerical Computation, by Fox, Pergamon Press, 1966, S.183-2oo.

Good I.J.: A Five-year Plan for automatic Chess, in Machine Intelligence 2, by Dale and Michie, Oliver and Boyd, 1968. S.89-118.

Euwe M.: Schach mit dem Computer, IBM Nachrichten 196, 1969.

Scott J.J.: A Chess-Playing Program, in Machine Intelligence 4, by Meltzer and Michie, 1969.

Wir wenden uns nun Schachendspielen zu und betrachten ein 4-Figurenspiel der Form wK + wF × sK + sF oder wF × sF.
Da wir u.a. das Spiel D × T - das praktisch stets für D gewonnen ist - behandeln wollen, kommt Auflisten nicht in Betracht. Somit ist die Verwendung des Algorithmus (I,II) von §8 gerechtfertigt und soll nun weiter diskutiert werden.

Symmetrisierung:

Das Schachbrett und -spiel hat, falls keine Rochade zugelassen wird, eine Symmetrie, und falls auch keine Bauern vorhanden sind - und dies ist der von uns betrachtete Fall - drei Symmetrien. Wir machen nur von zweien Gebrauch, indem wir eine Figur auf einen Quadranten beschränken. Dadurch reduziert sich die Zahl der Stellungen auf den vierten Teil. Die dritte Symmetrie nicht auszunutzen, hat folgenden Vorteil: Einmal hat man durch die Gleichheit der Anzahl symmetrischer Gewinn- bzw. Verluststellungen eine Kontrolle über den Algorithmus und über die Rechenanlage und zum anderen steigt der Programmier- und Zeitaufwand bei Beachtung der dritten Symmetrie erheblich. Die Symmetriesierung wirkt sich nämlich auf das Zugrecht aus. Wird der Bereich der symmetrisierten Figur durch diese überschritten, muß geeignet gespiegelt werden. Bei zwei Symmetrien kann nun durch entsprechende Brettnumerierung erreicht werden, daß die durch die Spiegelungen nötigen Maschinenoperationen kurz und schnell sind.

Komprimierung durch die Speicherstruktur einer Rechenanlage:

Zur Vereinfachung der Beschreibung gehen wir von der Annahme aus, daß wir eine 64-Bit-Maschine zur Verfügung haben. Ein Ganzwort ist für uns dann ein Element aus $\mathcal{L}_{64} := \mathcal{L}_{1,64}$, und wir nehmen weiter an, daß Elementaroperationen vel, et, $^-$ gegeben sind in der Art (Vektoroperationen auf $\mathcal{L}_{64}$):

Für $a = (a_1,..,a_{64})$ und $b = (b_1,..,b_{64})$ sei

$$a \text{ vel } b = (a_1 \vee b_1, ..., a_{64} \vee b_{64})$$

$$a \text{ et } b = (a_1 \wedge b_1, ..., a_{64} \wedge b_{64})$$

$$\bar{a} = (\bar{a}_1, ..., \bar{a}_{64}) \quad .$$

Weiter bedeute die Schreibweise $i \in a \;:\Leftrightarrow\; a_i = L$ und sei durch eine Prozedur realisiert.

Wir werden also eine Figur F bitweise verschlüsseln. Falls wir F so wählen können, daß der Algorithmus weitgehend unabhängig von der Stellung von F (durch simultane Operationen) arbeiten

kann, haben wir also nicht nur den Faktor 1/64 im Speicherbedarf, sondern mit Ausnahme der F betreffenden Operationen auch im Zeitbedarf. Allerdings konnte auf der verwendeten Rechenanlage nur ein Faktor 1/32 erhalten werden.
Die dadurch auftretenden Probleme liegen im wesentlichen bei der Beschreibung des Zugrechtes von F und der Berücksichtigung der Symmetrisierung. Während man das Zugrecht Γ der übrigen Figuren durch geeignete Listen angibt, wird das Zugrecht von F entsprechend der bitweisen Verschlüsselung durch einen Vektor Gamma[1 : 64] (Gamma[F] $\in \mathcal{B}_{64}$) beschrieben, gemäß der Erklärung

$$FN \in \Gamma F \quad \Leftrightarrow \quad FN \in \text{Gamma}[F] \quad ,$$

wobei Behinderungen durch andere Figuren nicht berücksichtigt sind. Die eventuell nötige Berücksichtigung anderer Figuren läßt sich durch wenige Elementaroperationen und Abprüfungen unter Verwendung von ähnlich wie Gamma aufgebauten Vektoren mit geringem Aufwand erledigen. Ebenso lassen sich die durch die Symmetrisierung erforderlichen zwei Spiegelungen durch Shiftoperationen einfach bewältigen, so daß sich hinsichtlich der F betreffenden Operationen der Zeitbedarf durch die bitweise Verschlüsselung und Verwendung von Gamma immerhin auf etwa 1/1o reduziert.

Blockeinteilung der Stellungen:

Trotz optimaler Speicherausnutzung der Rechenanlage durch bitweise Verschlüsselung ist die Zahl der benötigten Ganzworte noch zu groß, und wir werden die Stellungen in Blöcke mit einer festen Figur einteilen. Dazu stellen wir einige Anzahlen zusammen:

Gesamtzahl der Stellungen (mit illegalen, bei 2 Symmetrien)

$$16 \cdot 64^3 = 4\ 194\ 3o4$$

Zahl der entsprechenden Ganzworte

(*) $$16 \cdot 64^2 = 16 \cdot 4o96 = 65\ 536$$

Gesamtzahl der legalen Stellungen (bei 2 Symmetrien)

$$3\ 418\ 928\ .$$

Wir versuchen nun die symmetrisierte (SY), die bitweise verschlüsselte (BV) und die blockeinteilende (BL) Figur günstig zu wählen.
Klar ist, daß wir SY verschieden von BV annehmen.
Die Aufspaltung (*) legt nahe (da 4o96 für den Arbeitsspeicher eine geeignete Größe darstellt), daß SY gleich BL gewählt wird.
Um den Blockwechsel bei Spiegelung möglichst gering zu halten, werden wir SY so bestimmen, daß die durchschnittliche Zugmöglichkeit klein ist. Dies trifft auf den König zu.
Wir wählen SY = wK und somit auch BL = wK .
Bei der Festlegung von BV spielt eine Rolle, daß auf BV keine andere Figur (oder möglichst wenige Figuren) achtet, um nicht bitweise, sondern simultan und zellenweise zu arbeiten.
Dies ergibt gemäß §8,(1o) : BV = wF .

Wir haben also für festes wK ($1 \leq wK \leq 16$) Matrizen ${}^{wK}G_w^{(n)}$ und ${}^{wK}V_s^{(n)}$ über $\mathcal{B}_{64}$, die durch die schwarze Stellung s ($1 \leq s \leq 4o96$) indiziert sind und die wF ($1 \leq wF \leq 64$) im Bitmuster (${}^{wK}G_w^{(n)}[s] \in \mathcal{B}_{64}$) verschlüsselt haben.

Der Algorithmus (I,II) wurde entsprechend der vorstehenden Vereinbarungen in ALGOL unter Verwendung einiger code-Prozeduren programmiert. Wir geben eine knappe algorithmische Formulierung an:

GEWINN:
for all wK do
MITTELS WK: ${}^{wK}G_w^{(n)} := \underset{wKN \in \Gamma wK}{\mathrm{Vel}} \; {}^{wKN}V_s^{(n-1)}$

ERGAENZT MITTELS WF:
for all s do ${}^{wK}G_w^{(n)}[s] := \underset{wF \in {}^{wK}V_s^{(n-1)}[s]}{\mathrm{Vel}} \; \mathrm{Gamma}[wF]$

SODANN: ${}^{wK}G_w^{(n)} := {}^{wK}G_w^{(n)} \;\mathrm{et}\; {}^{wK}\overline{G}_w[n-1]$

VERLUST:
for all wK do
for all s do ${}^{wK}V_s^{(n)}[s] := \underset{sN \in \Gamma s}{\mathrm{ET}} \; {}^{wK}G_w[n][sN] \;\mathrm{et}\; {}^{wK}\overline{V}_s[n-1][s]$

if nicht fertig goto GEWINN ;

Als Kommentar begnügen wir uns mit folgendem:
Für die Rechnung ist es nur nötig die aktuellen (oder letzten) Matrizen $G_w^{[n]}$, $V_s^{[n]}$, $V_s^{(n)}$ gespeichert zu haben.
Γ und Gamma bedeuten das modifizierte Zugrecht, mit Ausnahme der in §8,(8) bis (1o) aufgezeigten Möglichkeiten. Bei GEWINN MITTELS WK muß gegebenenfalls auch noch gespiegelt werden. Die Rückrechnung im Sinne des §8 wird bei GEWINN MITTELS WF und (falls sinnvoll) bei VERLUST vorgenommen. Bei GEWINN MITTELS WF muß geeignet bitweise gearbeitet werden.

Andere Algorithmen zur Behandlung von Schachendspielen sind bei D.E. Knuth, The Art of Computer Programming, Vol.1, Seite 262 - 27o und bei R. Bellman in Applied Combinatorial Mathematics (by E.F. Beckenbach), Seite 233, 234 vorgeschlagen und angedeutet. Eine allgemeine Behandlung von Endspielen mit vier (oder mehr) Figuren läßt sich jedoch wegen des auftretenden Speicher- und Zeitbedarfs nur bei sparsamster, also bitweiser Verschlüsselung und mit daran in der beschriebenen Weise angepaßten Methoden praktisch durchführen.

Die Rechnungen wurden an einer TR4 von AEG-Telefunken des Leibnizrechenzentrums der Bayerischen Akademie der Wissenschaften, München durchgeführt. Die Anlage stellt dem Benutzer einen Kernspeicher von ca. 19 ooo Ganzworten zu 48 Bit sowie einen Plattenspeicher und mehrere Bandgeräte zu Verfügung. Die Ausführungszeiten betragen für elementare Internoperationen ca. 8 μsec.
Wir stellen die Ergebnisse, die für einige Schachendspiele gewonnen wurden, kurz zusammen. Dabei betrachten wir das Spiel stets als eines mit w am Zug und erinnern an D4.4 über die Zugzahl eines Spieles.

1. Turm-Endspiel wK + wT × sK :

Es handelt sich um ein 16-zügiges Spiel, das stets gewonnen ist. 16-zügige Gewinnstellungen gibt es insgesamt 916, von denen aber

nur 121 wesentlich verschieden (d.h. nicht durch Drehung oder Spiegelungen ineinander überführbar) sind. Zum Beispiel
w Ka1 Tb2, s Kc3 und w Ka1 Th8, s Kd5.

2. Dame-Endspiel wK + wD × sK :

Es erweist sich als ein stets gewonnenes, 1o-zügiges Spiel mit kurioserweise nur einer wesentlich verschiedenen längstzügigen Stellung. (Insgesamt gibt es 8 solcher Stellungen). Diese Extremstellung ist w Ka1 Db2, s Ke6.

Bei einem 3-Figuren-Spiel ist es natürlich für einen Schachspieler kein Problem, den Gewinn zu realisieren, aber es ist schwierig, ihn in der optimalen Zugzahl zu erreichen. In diesem Sinne wird von Verfassern von Büchern über Endspieltheorie die maximale Zugzahl nur teilweise richtig angegeben und eine zugehörige Belegstellung fehlt zumeist. Im Gegensatz dazu ist es bei einem 4-Figuren-Spiel oft schon schwierig überhaupt richtig zu spielen und zu gewinnen.Bezüglich der Bedeutung der Zugzahl erinnern wir an die Vereinbarung von S8.3, die aus Gründen der Zweckmäßigkeit noch modifiziert wird: Die Zugzahl gibt (bei bester Verteidigung) die Anzahl der Züge bis zum Schlagen des sK - also einen Zug mehr als bis zum Matt - oder bis zur Eroberung der sF an.
Die angeführte Rechenzeit schließt den Druck einer einfachen Ergebnisliste ein.

3. T × L:

Das Schachspiel Turm gegen Läufer endet meistens Remis. Dennoch ist der mögliche Gewinn in den langwierigsten Fällen erst in 18 Zügen zu verwirklichen. Es gibt insgesamt 224 solcher Stellungen, wovon nur 28 wesentlich verschieden sind. Zum Beispiel
w Ka4 Tc3 , s Ka7 La6 . Rechenzeit ca. 6 Stunden und 3o Minuten.

4. T × S:

Beim Turm gegen Springer-Spiel ist ebenfalls in der Mehrzahl der Stellungen Remis (aber nicht in fast allen wie bei T × L).

Es ist ein 27-zügiges Spiel mit 2 wesentlich verschiedenen längstzügigen Stellungen (insgesamt 16). Für eine solche Stellung geben wir eine mögliche Partie an:
w Kd1 Th1 , s Kb1 Sg4. 1, Th4! Se5! 2, Te4! Sf7! 3, Tb4+ (auch Kd2) Ka2(!) 4, Kc2 (auch Tb6) Ka3(!) 5, Kc3 (auch Tb6) Sd6! 6, Tb6! Se4+! 7, Kd3!! Sf2+! 8, Kc4! Sd1! 9, Tb3+! Kd1! 1o, Tf3 (auch Tg3,Th3) Sb2+! 11, Kc3! Ka3! 12, Tg3 (Tempozug!, auch Th3) Sa4+! 13, Kc4! Ka2! 14, Kb4 (auch Tg5) Sb2! 15, Tg4! Sd3+! 16, Kc3! Sc5! 17, Tc4! Se6! 18, Ta4+! Kb1(!) 19, Ta5! Sg7! 2o, Te5! Nun ist der Springer eingeschlossen und kann durch Annäherung des wK erobert werden, wobei lediglich auf das Tempomanöver in 23.geachtet werden muß. Ka2
21, Kd4 Kb3 22, Kd5 Kc3 23, Kc6! Kd4 24, Kd6 Kd3 25, Ke7 Kd4 26, Tg5 und gewonnen mit Eroberung des Springers.
Rechenzeit ca. 14 Stunden und 16 Minuten.

5. D × T:

Das Dame gegen Turm-Spiel ist ein 31-zügiges Spiel, das bis auf einige Remisstellungen (in denen die verteidigende Partei ihre Pattsetzung erzwingt oder mit Hilfe von Pattdrohungen eine Zugwiederholung erreicht) stets gewonnen ist. Es gibt 4 wesentlich verschiedene längstzügige Stellungen (insgesamt 32), zum Beispiel w Ka2 Da3 , s Ke4 Th2 .
Rechenzeit ca. 29 Stunden und 9 Minuten.

Um die Möglichkeit zu erörtern, den Algorithmus in vorstehender Form auf andere Spiele anzuwenden, stellen wir nochmals die wesentlichen Daten über die behandelten Spiele zusammen:

Spiel	Zugzahl	Rechenzeit gesamt	pro Zug [sec]	pro Zug u.Stellung[μsec]
1. T	16	9Min	33,8	516
2. D	1o	6,5Min	39	595
3. T×L	18	6Std.3oMin	1296	3o9
4. T×S	27	14Std.16Min	19o2	454
5. D×T	31	29Std. 9Min	3384	8o8

Dabei liegt als Zahl der betrachteten Stellungen $1/4 \cdot 64^3 = 2^{16}$ bzw. $1/4 \cdot 64^4 = 2^{22}$ zugrunde. Beim Vergleich der Rechenzeit pro Zug und Stellung ist zu berücksichtigen, daß durch die Rückrechnungs-Steuerung des Algorithmus gegebenenfalls pro Zug nicht alle Stellungen behandelt werden, sowie daß die Figuren sich in der Schwierigkeit ihrer Gangart unterscheiden.

Auch unter der Annahme, daß ein schnellerer Rechner zur Verfügung steht, kann man wohl - je nach Zugzahl - 2^{25} bis 2^{28} als obere Grenze für die Zahl der Positionen eines Spieles ansehen, das im angegebenen Verfahren analysiert werden kann. In diesem Sinne führen wir folgende Abschätzung für die Anzahl der Positionen einer Auswahl von Spielen an.

Schach mit 5 Figuren	$1/4 \cdot 64^5 = 2^{28}$
Dame mit 6 Figuren	$1/4 \cdot 32^6 = 2^{28}$
"Kleine Mühle" (auf 3×3 Brett)	$\approx 9^6 \approx 2^{19}$
Mühle mit 6 Figuren	$1/8 \cdot 24^6 \approx 2^{25}$

Setzspiele, bei denen die Bedeutung in der Anzahl der Figuren liegt, sind praktisch unzugänglich. So hat z.B. Hex auf dem 11×11 Brett ca. $3^{121} \approx 2^{194}$ Stellungen. Aber auch bei Berücksichtigung nur legaler Stellungen und Beschränkung auf den kleinsten interessierenden Fall, etwa ein 7×7 Brett mit 14 Steinen, hat man ca. $49^{14}/(7!)^2 \approx 2^{53}$ Stellungen.

Literaturverzeichnis

[1] Berge,C.: The Theory of Graphs. J.Wiley Inc.

[2] Birkhoff,G.: Lattice Theory. AMS Colloq.Publ.XXV.

[3] Gale,D. and Stewart,F.M.: Infinite Games with Perfect Information. Annals of Math.Studies 28,245-266 (1953).

[4] Gericke,H.: Theorie der Verbände.

[5] Give'on,Y.: Lattice Matrices. Inform.and Control 7, 477-484 (1964).

[6] Luce,R.C.: A Note on Boolean Matrix Theory. Proc.AMS 3, 382-388 (1952).

[7] Maghout,K.: Applications de l'Algèbre de Boole à la théorie des graphes et aux programmes linéaires et quadratiques. Cahiers du Centre d'Etude de Recherche Opérationelle 5, 21-99 (1963).

[8] Neumann,J.von and Morgenstern,O.: Theory of Games and Economic Behavior. Princeton, 1944.

[9] Steinhaus,H.: Definitions for a Theory of Games and Pursuit (1925). Nachdruck in: Naval Res.Logist.Quart.7,1o5-1o8(196o).

Ferner finden sich einige Hinweise auf Literatur bezüglich "Schachspielender Computer" auf Seite 56.