

Seminararbeit  
Multiagentensysteme: Entscheidungsfindung für  
den individuellen Vorteil

von Onno Bergob  
im Rahmen des Informatik-Seminars Methoden der KI  
bei Prof. Dr. Sebastian Iwanowski erstellt

letzte Änderung 16. Juni 2010

# Inhaltsverzeichnis

<b>1</b>	<b>Einordnung</b>	<b>3</b>
<b>2</b>	<b>Eigenschaften Agent</b>	<b>3</b>
<b>3</b>	<b>Nutzen und Präferenzen</b>	<b>3</b>
<b>4</b>	<b>Zustandsübergang</b>	<b>4</b>
4.1	Payoff-Matrix . . . . .	5
<b>5</b>	<b>Spiele in Normalform</b>	<b>5</b>
5.1	Definition Normalformspiele . . . . .	6
<b>6</b>	<b>Lösungskonzepte und Lösungseigenschaften</b>	<b>7</b>
6.1	Beste Antwort . . . . .	7
6.2	Dominante Strategien . . . . .	7
6.3	Nash-Gleichgewicht . . . . .	8
6.3.1	Striktes und schwaches Nash-Gleichgewicht . . . . .	9
6.3.2	Reine Strategien und gemischte Strategien . . . . .	9
6.3.3	Erwarteter Nutzen . . . . .	10
6.3.4	Beispiel Nash-Gleichgewicht . . . . .	10
6.3.5	Berechnung Nash-Gleichgewicht an Beispiel . . . . .	10
6.3.6	Nash Theorem . . . . .	11
6.4	Pareto Effizienz . . . . .	12
6.5	Maximierung des Gemeinwohls . . . . .	12
<b>7</b>	<b>Konstantsummenspiel</b>	<b>12</b>
7.1	Nullsummenspiel . . . . .	12
<b>8</b>	<b>The shadow of the future</b>	<b>13</b>
<b>9</b>	<b>Programmgleichgewicht</b>	<b>13</b>
<b>10</b>	<b>Umsetzung Multiagenten Szenarien</b>	<b>13</b>
<b>I</b>	<b>Anhang</b>	<b>16</b>

## 1 Einordnung

Die wissenschaftliche Betrachtung von Multiagentensystemen mit dem Augenmerk auf die Entscheidungsfindung (für den individuellen Vorteil) ist ein Teilgebiet der Spieltheorie. Die Interaktion von selbstständigen Agenten für den *individuellen Vorteil* (self-interested) ist dem Unterbereich der *nicht-kooperativen* Spiele zuzuordnen.

## 2 Eigenschaften Agent

Die Basiseinheit der Modellierung ist bei der nicht-kooperativen Spieltheorie der individuelle Agent mit seinen Präferenzen und möglichen Aktionen.

Bei den betrachteten Modellen wird angenommen, dass der Entscheider (Agent) *rational* in dem Sinne ist, dass er Alternativen kennt, Erwartungen über Unbekanntes treffen kann, klare Präferenzen hat, und sein Handeln frei wählen kann.

Entscheidungsfindung für den individuellen Vorteil heißt nicht, dass die Agenten unbedingt anderen Agenten Schaden zufügen wollen. Stattdessen hat ein Agent eine eigene Vorstellung davon, welche Zustände der Welt er gut findet. Das bedeutet für seine Aktionen, dass er versucht die Welt in diese Zustände zu bringen. Dabei wird angenommen, dass die bevorzugten Zustände vom Besitzer des Agenten bestimmt werden.

## 3 Nutzen und Präferenzen

Im Folgenden wird der Einfachheit halber immer von zwei Agenten ( $i$  und  $j$ ) geschrieben, wenn Situationen mit Multiagenten beschrieben werden. Der dominierende Ansatz die Präferenzen eines Agenten zu modellieren ist die Nutzentheorie nach von Neumann und Morgenstern (UTILITY THEORY). Dabei wird angenommen, dass es eine Menge  $\Omega = \{\omega_1, \omega_2, \dots\}$  an Zuständen gibt. Die Präferenzen eines Agenten werden formal durch eine sogenannte Nutzenfunktion (von Neumann-Morgenstern UTILITY FUNCTION), von der jeder Agent eine eigene hat, beschrieben. Dabei ordnet die Funktion einem Zustand eine reelle Zahl<sup>1</sup> zu, die angibt, wie gut der Zustand für den Agenten ist (bzw. wie groß der Nutzen). Je größer die Zahl, desto besser ist das Ergebnis für den Agenten. Für das Verhalten des Agenten bedeutet das, dass er versucht seinen Nutzen zu maximieren durch Bevorzugung (Präferenz) von bestimmten Zuständen. Für den Agenten  $i$  ist der Nutzen eines Zustands durch die Funktion

$$u_i : \Omega \rightarrow \mathbb{R}$$

abgebildet und für den zweiten Agenten  $j$  durch die Funktion

$$u_j : \Omega \rightarrow \mathbb{R}.$$

---

<sup>1</sup>Von Neumann und Morgenstern haben nachgewiesen, dass die Größenordnung der Funktionswerte der Nutzenfunktion für Zustände unwichtig ist. Siehe [SLB09, Seite 54]

Die Nutzenfunktion führt zu einer Sortierung (Präferenzordnung) der Zustände nach den Ergebnissen der Funktion. Angenommen es gibt zwei Zustände  $\omega_1$  und  $\omega_2$  in  $\Omega$  und  $u_i(\omega_2) \geq u_i(\omega_1)$ , dann würde Agent  $i$  den Zustand  $\omega_2$  mindestens genauso gut (oder besser) finden, wie Zustand  $\omega_1$ . Man kann diese Ordnung der Zustände durch die Nutzenfunktion auch folgendermaßen verkürzt beschreiben:

$$\omega_2 \succeq_i \omega_1$$

für

$$u_i(\omega_2) \geq u_i(\omega_1).$$

Ein Beispiel für eine konkrete Nutzenfunktion folgt im nächsten Abschnitt.

## 4 Zustandsübergang

Der Wechsel zu einem bestimmten Zustand wird durch die Aktionen aller Agenten bestimmt. Dieses Aktionentupel (eine Aktion pro Agent) wird auch Aktionsprofil (action profile) genannt.

Die Zustandsübergangsfunktion ist:

$$\tau : Ac_i \times Ac_j \rightarrow \Omega,$$

wobei  $Ac_i$  für die Aktion des Agenten  $i$  steht. Zur Vereinfachung wird angenommen, dass jeder Agent nur die Wahl aus zwei verschiedenen Aktionen hat: DEFECT (Zusammenarbeit verlassen, abtrünnig werden) und COOPERATE (zusammenarbeiten). Dazu ein Beispiel für eine konkrete Zustandsübergangsfunktion:

$$\tau(D, D) = \omega_1, \quad \tau(D, C) = \omega_2, \quad \tau(C, D) = \omega_3, \quad \tau(C, C) = \omega_4.$$

Hier ist der Fall gezeigt, bei dem jede unterschiedliche Kombination von Aktionen auf einen anderen Zustand abgebildet ist. Das bedeutet, dass in diesem Fall die Aktionen beider Agenten entscheidend sind für den neuen Zustand. Um das Beispiel konkreter zu machen, muss noch die Abbildung der Nutzen der Zustände für die beiden Agenten  $i$  und  $j$  ergänzt werden:

$$u_i(\omega_1) = 1, \quad u_i(\omega_2) = 1, \quad u_i(\omega_3) = 4, \quad u_i(\omega_4) = 4$$

$$u_j(\omega_1) = 1, \quad u_j(\omega_2) = 4, \quad u_j(\omega_3) = 1, \quad u_j(\omega_4) = 4.$$

Nun kann man die Präferenzrelationen darstellen:

$$(C, C) \succeq_i (C, D) \succ_i (D, C) \succeq_i (D, D)$$

$$(C, C) \succeq_j (D, C) \succ_j (C, D) \succeq_j (D, D).$$

Das erste Zeichen entspricht der Aktion des Agenten  $i$ , das zweite Zeichen der Aktion von Agent  $j$ . Daraus lässt sich für Agent  $i$  schlussfolgern, dass er zusammenarbeiten (C) sollte, da alle Zustände, in denen er zusammenarbeitet einen

größeren Nutzen für ihn haben als alle Zustände, in denen er die Zusammenarbeit verweigert. Für Agent  $j$  ist das für dieses Szenario auch der Fall. Durch diese Konstellation ergibt sich der Fall, dass das Handeln der Agenten unabhängig voneinander gewählt werden kann. Die *rationale* Entscheidung beider Agenten ist Zusammenarbeit.

#### 4.1 Payoff-Matrix

Das Interaktionsszenario lässt sich noch auf eine andere, weit verbreitete Weise darstellen als *Payoff* ( $\cong$  UTILITY)-*Matrix*:

	$j$ defects	$j$ cooperates
$i$ defects	(1, 4)	(4, 1)
$i$ cooperates	(1, 4)	(1, 1)

Tabelle 1: Beispiel für eine Payoff-Matrix

Eine Zeile in der Tabelle steht für eine Aktion des Agenten  $i$ , eine Spalte steht für eine Aktion des Agenten  $j$ . Jede Zelle der Tabelle stellt einen Zustand des Spiels dar. Die Zelle mit dem Tupel (1, 1) beispielsweise repräsentiert den Zustand, der eintritt, wenn Agenten  $i$  und  $j$  jeweils die Aktion cooperate durchführen. Die erste Zahl des Tupels in jeder Zelle gibt den Nutzen des Zustandes für den Agenten  $i$  und die zweite Zahl den Nutzen für den Agenten  $j$  wieder. Mit einer Payoff-Matrix kann man ein Spiel in Normalform vollständig beschreiben.

## 5 Spiele in Normalform

Agenten wollen den Nutzen maximieren. Das ist aber nicht mehr ganz so trivial, wenn man die Situation mit zwei Agenten betrachtet, die *beide* ihren Nutzen maximieren wollen.

Dazu ein Beispiel, was in der Literatur oft als PRISONER'S DILEMMA GAME wiedergegeben wird: Es gibt zwei Personen, Person  $i$  und Person  $j$ , die eines gemeinsam verübten Verbrechens verdächtigt werden. Sie werden getrennt voneinander verhört und können nicht miteinander kommunizieren. Nun wird ihnen folgendes gesagt: Wenn  $i$  erzählt, dass  $j$  die Tat begangen hat (defect), und  $j$  wiederum  $i$  nichts sagt (cooperate), dann kommt  $i$  frei und  $j$  muss für drei Jahre ins Gefängnis. Das ganze gilt auch umgekehrt. Zudem wird ihnen erzählt: Wenn  $i$  und  $j$  beide nichts sagen (cooperate), kommen sie beide für ein Jahr ins Gefängnis. Wenn  $i$  und  $j$  sich gegenseitig beschuldigen (defect), kommen beide für zwei Jahre ins Gefängnis.

Hier noch einmal das Szenario als Payoff-Matrix:

	$j$ defects	$j$ cooperates
$i$ defects	$(-2, -2)$	$(0, -3)$
$i$ cooperates	$(-3, 0)$	$(-1, -1)$

Tabelle 2: Payoff-Matrix des PRISONER'S DILEMMA GAME

Dabei stehen die Nutzenwerte für verlorene Jahre durch Gefängnisstrafe ( $-3$  z. B. bedeutet eine Strafe von drei Jahren Gefängnis).

Nun ist die Frage was man aus der Sicht einer der Verdächtigen tun sollte. Die Schlussfolgerung eines rationalen Agenten sieht folgendermaßen aus: Die beste Aktion auf die Aktion defect ist defect. Die beste Aktion (maximaler Nutzen), wenn der andere Verdächtige cooperate wählt, ist defect. Das bedeutet, dass rational betrachtet die beste Aktion defect ist. Wenn nun aber zwei Agenten beide danach handeln entsteht der Zustand  $(D, D)$  und beide müssen für zwei Jahre ins Gefängnis. Nun wäre das aber nicht notwendig, wenn man sich überlegt, dass sie auch zusammenarbeiten könnten und so nur jeder für ein Jahr ins Gefängnis müsste und dadurch beide ihren eigenen Nutzen erhöhen könnten. Wenn aber einer der Verdächtigen weiß, dass der andere zusammenarbeiten will, ist die beste Aktion ihn zu verraten. Rational betrachtet ist also defect die logische Entscheidung, obwohl durch den Zustand  $(D, D)$  Nutzen sozusagen verschwendet wird. Genau wegen dieses Dilemmas um den verschwendeten Nutzen heißt das Szenario so wie es heißt.

Die allgemeine Form dieses Szenarios ist in Tabelle 3 abgebildet.

	$j$ Aktion1	$j$ Aktion2
$i$ Aktion1	$(a, a)$	$(b, c)$
$i$ Aktion2	$(c, b)$	$(d, d)$

Tabelle 3: Jedes Szenario mit  $b > d > a > c$  entspricht dem Prisoner's Dilemma

## 5.1 Definition Normalformspiele

Für Spiele in Normalform gilt:

- Alle Agenten kennen alle möglichen Zustände und auch alle möglichen Nutzen für alle Aktionen aller Agenten. Diese Eigenschaft wird vollständige Information genannt.
- Der Wechsel zu einem bestimmten Zustand wird allein durch die Aktionen aller Agenten (Aktionentupel der ausgeführten Aktionen) bestimmt. Das heißt, dass keine anderen Faktoren, wie die Umwelt, die Zeit, eine Rolle spielen. Andere Spielformen, wie die Bayesschen Spiel (Zustand auch abhängig von Zufälligkeit in der Umwelt der Agenten) oder Extensive Spiele (Zustand auch abhängig vom Zeitpunkt) lassen sich auf die fundamentale Form der Normalformspiele reduzieren.
- Die Agenten führen ihre Aktionen gleichzeitig durch.

- Ein Agent weiß nicht, welche Aktionen die anderen Agenten zu der gleichen Zeit durchführt.

## 6 Lösungskonzepte und Lösungseigenschaften

### 6.1 Beste Antwort

Die sogenannte *beste Antwort* (best response) ist diejenige Strategie (Aktion), die bei einer bestimmten Strategienkonstellation der anderen Spieler den größtmöglichen Nutzen erbringt. Dazu ein Beispiel:

	<i>j</i> cooperates	<i>j</i> defects
<i>i</i> cooperates	(4, 4)	(1, 4)
<i>i</i> defects	(1, 4)	(4, 1)

Tabelle 4: Beste Antwort

Wenn Agent *j* die Aktion cooperate durchführt (hellgraue Spalte), dann hat Agent *i* den größten Nutzen (blaue Farbe) durch die Aktion cooperate. Demnach ist für Agent *i* cooperate die beste Antwort auf die Aktion cooperate von Agent *j*. Für die Aktion defect des Agenten *j* (dunkelgraue Spalte) ist die Situation umgekehrt: Die Aktion defect bringt ihm mehr Nutzen als die Aktion cooperate. Demnach ist für Agent *i* defect die beste Antwort auf die Aktion defect von Agent *j*.

Im allgemeinen Fall kann es auch mehrere beste Antworten geben. Man muss sich nur ein Szenario vorstellen in dem zwei verschiedene Aktionen in einer bestimmten Strategienkonstellation der anderen Spieler den gleichen Nutzen erzielen würden, und das gleichzeitig der größtmögliche Nutzen wäre.

Da man im allgemeinen nicht davon ausgehen kann, die Aktionen der anderen Agenten zu kennen, ist die Idee der besten Antwort kein Lösungskonzept, dass einem Agenten bei der Wahl der Aktion unmittelbar weiterhelfen kann. Dafür bildet die beste Antwort eine Grundlage für die anschließenden Lösungskonzepte.

### 6.2 Dominante Strategien

Eine *dominante Strategie* ist die beste Antwort auf alle Strategien der anderen Spieler. Anhand des durch die Payoff-Matrix (Tabelle 5) definierten Szenarios lässt sich das folgendermaßen erklären: Wenn der Agent *j* die Strategie defect wählt, bekommt Agent *i* den größtmöglichen Nutzen durch die Aktion cooperate. Da das gleiche auch für die Aktion cooperate von Agent *j* zutrifft, ist die Strategie cooperate für Agent *i* die *beste Antwort* auf *alle* Strategien des anderen Spielers und somit eine dominante Strategie. In der Tabelle 5 sind die Nutzenwerte, die dabei *pro Spalte* verglichen werden, blau (höherer Nutzen) und rot (niedriger Nutzen) eingefärbt.

	$j$ cooperates	$j$ defects
$i$ cooperates	(4, 4)	(4, 1)
$i$ defects	(1, 4)	(1, 1)

Tabelle 5: Dominante Strategie

### 6.3 Nash-Gleichgewicht

Das Nash-Gleichgewicht beschreibt eine Eigenschaft eines Zustandes. Ein Zustand ist im Nash-Gleichgewicht, wenn kein größerer Nutzen erreicht werden kann durch die Änderung genau *einer* Aktion. Man kann auch sagen, Nash-Gleichgewicht bedeutet, dass jeder Agent die beste Antwort auf die Aktionen der anderen Agenten gegeben hat. Das bedeutet für jeden einzelnen Agenten (- und genau der ist Mittelpunkt dieser Arbeit), dass es für ihn nicht von Vorteil ist, eine andere Aktion zu wählen.

Dazu ein Beispielszenario aus dem Bereich Straßenverkehr: Zwei Fahrer fahren mit ihren Autos auf einer Straße in einem Land ohne Verkehrsregeln aufeinander zu. Beide müssen selbstständig entscheiden, auf welcher Seite sie fahren wollen. Wenn beide Fahrer von sich aus gesehen die selbe Seite wählen (links oder rechts), haben beide einen guten Nutzen (fahren aneinander vorbei). Tun sie das nicht, haben sie einen niedrigeren Nutzen (einen Unfall). Die Spiel-Matrix ist in Tabelle 6 dargestellt.

	$j$ links	$j$ rechts
$i$ links	(1, 1)	(0, 0)
$i$ rechts	(0, 0)	(1, 1)

Tabelle 6: Nash-Gleichgewicht beim Straßenverkehr

Ein Beispiel dafür, dass nicht jedes Szenario ein Nash-Gleichgewicht hat, ist das "matching pennies"-Spiel. Zwei Spieler wählen unabhängig voneinander gleichzeitig eine Seite einer Münze. Wenn sie die gleiche Seite wählen, gewinnt Spieler  $i$ , wenn sie unterschiedliche Seiten wählen gewinnt Spieler  $j$ . Das bedeutet, es gibt keinen Zustand in dem beide zufrieden sind und sich nicht einer der Spieler wünscht, die andere Münzseite gewählt zu haben. Somit gibt es in diesem Szenario keinen Zustand der im reinen Strategie Nash-Gleichgewicht ist. Was das *reine* Strategie Nash Gleichgewicht ist und welche andere Form es noch gibt, wird noch erläutert. Dargestellt ist das Szenario "matching pennies" in der Tabelle 7.

	$j$ Kopf	$j$ Zahl
$i$ Kopf	(1, -1)	(-1, 1)
$i$ Zahl	(-1, 1)	(1, -1)

Tabelle 7: Matching pennies

### 6.3.1 Striktes und schwaches Nash-Gleichgewicht

Man kann das Nash-Gleichgewicht noch unterteilen in ein *schwaches* und ein *striktes* Nash-Gleichgewicht. Für ein striktes Nash-Gleichgewicht gilt für jeden Agenten, dass er nur einen einzigen beste Antwort auf die Strategien der anderen Agenten hat. Bei dem schwachen Nash-Gleichgewicht kann er auch mehrere beste Antworten haben. Das schwache Nash-Gleichgewicht ist weniger stabil, denn es gibt mindestens einen Agenten, der ohne Einbuße von Nutzen eine andere Strategie, als die Strategie, die im Nash-Gleichgewicht ist, wählen könnte. Um das zu Veranschaulichen, kann man die Payoff-Matrix in Tabelle 8 betrachten. Das blau gekennzeichnete Nutzentupel (Zustand im Nash-Gleichgewicht) gibt dem Agenten  $j$  einen Nutzen von 1 für die Aktion links als Antwort auf die Aktion links des Agenten  $i$ . Da Agent  $j$  aber für die Aktion rechts in der selben Situation einen genauso großen Nutzen bekommt, gibt es zwei beste Antworten. Der Agent könnte, ohne Nachteil hinsichtlich seines Nutzens, den Zustand, der nicht im Nash-Gleichgewicht ist, bevorzugen.

	$j$ links	$j$ rechts
$i$ links	(1, 1)	(0, 1)
$i$ rechts	(0, 0)	(1, 0)

Tabelle 8: Schwaches Nash-Gleichgewicht

### 6.3.2 Reine Strategien und gemischte Strategien

Strategien sind bis hierhin die Wahl und Ausführung einer Aktion für jede Situation (Aktionen der anderen Agenten). Diese Art der Strategie nennt sich *reine Strategie* (pure strategy). Die Berechnung, welche Zustände im reinen Strategie Nash-Gleichgewicht sind kann man folgendermaßen durchführen: Für jeden Zustand muss geprüft werden, ob das dazugehörige Aktionsprofil (Aktionentupel) die *beste Antwort* für *jeden* Agenten bedeutet. Um alle möglichen Nash-Gleichgewichte eines Szenarios zu finden bedeutet das für  $m$  Agenten und  $n$  Strategien insgesamt  $n^m$  Aktionentupel, für die geprüft werden muss, ob sie für alle Agenten die beste Antwort sind.

Eine andere Art der Strategie ist es, mit Hilfe einer Wahrscheinlichkeitsverteilung über jede reine Strategie eine reine Strategie zufällig auszuwählen. Diese Art der Strategie wird *gemischte Strategie* (mixed strategy) genannt. Dabei wird jedem Zustand zusätzlich zu dem Nutzen eine Wahrscheinlichkeit zugeordnet, die beschreibt, mit welcher Wahrscheinlichkeit der Zustand eintritt. Man kann eine reine Strategie als gemischte Strategie betrachten, bei der genau eine reine Strategie mit der Wahrscheinlichkeit 1 gespielt wird und alle anderen mit Wahrscheinlichkeit 0.

### 6.3.3 Erwarteter Nutzen

Der erwartete Nutzen ist der durchschnittliche Nutzen der Zustände, gewichtet mit den Wahrscheinlichkeiten.

Dazu ein Beispiel: Uwe und Peter haben zwei Möglichkeiten ihre Freizeit zu verbringen, nämlich zum Mars fliegen und ins Kino gehen. Wenn Uwe alleine seine Zeit verbringt hat er einen Nutzen von 100 für zum Mars fliegen und 50 für ins Kino gehen. Wenn Peter mit Uwe seine Freizeit verbringt, erhöht sich Uwe's Nutzen um den Faktor 1,8, egal welche Aktivität sie durchführen. Peter geht mit einer Wahrscheinlichkeit von 80% ins Kino und mit 20% Wahrscheinlichkeit fliegt er zum Mars. Nun lässt sich für Uwe der erwartete Nutzen für das Marsfliegen berechnen:  $0,2 * (1,8 * 100) + 0,8 * (50) = 76$ . Genauso wird der erwartete Nutzen für Uwe für den Kinogang berechnet:  $0,2 * (100) + 0,8 * (1,8 * 50) = 92$ . Demnach bevorzugt Uwe ins Kino zu gehen, weil er dadurch einen größeren erwarteten Nutzen hat.

### 6.3.4 Beispiel Nash-Gleichgewicht

Bei dem Spiel "Stein, Schere, Papier" (Payoff-Matrix in Tabelle 9) gibt es kein Nash-Gleichgewicht für reine Strategien. Das heißt, es gibt keinen Zustand bei dem ein Spieler nicht mehr Nutzen hätte durch eine andere reine Strategie. Wenn beide Spieler mit gleichverteilter Wahrscheinlichkeit ( $\frac{1}{3}$ ) Stein, Schere und Papier spielen ergibt sich ein Nash-Gleichgewicht. Wenn einer der Spieler diese gemischte Strategie anwendet, ist es für den Gegner egal, welche Strategie er anwendet. Bei jeder anderen Strategie kann der Gegner eine Strategie wählen, die ihm einen größeren Nutzen bringt und demnach zum Nachteil des Spielers ist. Wenn also einer der Spieler nicht gleichverteilt zufällig wählt, wird argumentiert<sup>2</sup>, dass die Möglichkeit besteht, dass die Strategie entdeckt und ausgenutzt wird.

	$j$ Stein	$j$ Schere	$j$ Papier
$i$ Stein	(0, 0)	(1, -1)	(-1, 1)
$i$ Schere	(-1, 1)	(0, 0)	(1, -1)
$i$ Papier	(1, -1)	(-1, 1)	(0, 0)

Tabelle 9: Stein, Schere, Papier

### 6.3.5 Berechnung Nash-Gleichgewicht an Beispiel

Nun noch ein Beispiel wie sich ein Nash-Gleichgewicht für gemischte Strategien ausrechnen lässt. Grundlage ist das schon vorgestellte "matching pennies"-Spiel. An der Payoff-Matrix (siehe Tabelle 10) lässt sich schnell sehen, dass kein reines Strategie Nash-Gleichgewicht existiert. Nun wird angenommen, dass beide Spieler zufällig wählen, und zwar wählt Spieler  $i$  mit der Wahrscheinlichkeit  $p$  Kopf und mit der Wahrscheinlichkeit  $1 - p$  Zahl. Damit nun Spieler  $j$  auch eine

<sup>2</sup>Nachzulesen u.a. in [SLB09, Seite 64]

gemischte Strategie anwendet, muss es ihm gleichgültig sein, welche Strategie er wählt. Nun kommt der erwartete Nutzen zum Einsatz. Der erwartete Nutzen der Strategie Kopf wird mit dem erwarteten Nutzen der Strategie Zahl aus den eben erläuterten Gründen gleichgesetzt (beide erwarteten Nutzen müssen gleich groß sein). Das kann man folgendermaßen notieren:

$$u_i(\text{Kopf}) = u_i(\text{Zahl})$$

$$1 * p + (-1) * (1 - p) = -1 * p + 1 * (1 - p)$$

$$p = \frac{1}{2}$$

Das bedeutet für das Beispiel, dass es für Spieler  $j$  egal ist (es gibt keine Strategie, die seinen erwarteten Nutzen erhöht), welche Aktion er wählt, wenn der Spieler  $i$  mit jeweils der Wahrscheinlichkeit  $\frac{1}{2}$  Kopf und Zahl spielt. Mit der gleichen Berechnung kann man zeigen, dass der Spieler  $i$  gleichgültig handeln kann, wenn Spieler  $j$  mit der Wahrscheinlichkeit  $\frac{1}{2}$  beide Möglichkeiten spielt. Nun ergibt sich ein schwaches Nash-Gleichgewicht: Beide Spieler machen den anderen Spieler gleichgültig in der Wahl und damit geben sie jeweils die *beste Antwort*. Ein Nash-Gleichgewicht in *gemischten Strategien* kann immer nur ein *schwaches Nash-Gleichgewicht* sein. In diesem Beispiel wird das durch die Gleichgültigkeit der Wahl deutlich: Kein Agent hat einen Vorteil aus einer anderen Strategie (als die im Nash-Gleichgewicht), andererseits ist jede andere Strategie für den einen Agenten genau so gut.

	$j$ Kopf	$j$ Zahl
$i$ Kopf	(1, -1)	(-1, 1)
$i$ Zahl	(-1, 1)	(1, -1)

Tabelle 10: Matching pennies

### 6.3.6 Nash Theorem

John Forbes Nash Jr., geboren 1928, promovierte 1950 an der Princeton University mit seiner Arbeit “Noncooperative Games”. Diese Arbeit enthielt eine These für die er letztendlich 1994 den Nobelpreis in Wirtschaftswissenschaften erhielt. Das Ergebnis seiner Untersuchungen war der Nachweis für den folgende Satz: “Jedes Spiel in dem jeder Spieler endlich viele möglichen Strategien hat, hat ein Nash-Gleichgewicht in gemischten Strategien”. Beweise für die *Komplexität des Findens eines Nash-Gleichgewichts* waren lange Zeit auf Nash-Gleichgewichte mit bestimmten Eigenschaften und bestimmte Spielformen beschränkt, bis 2006 die ersten Beweise für die seltene *Komplexitätsklasse PPAD-complete* durch Dasgkalis (für vier-Spieler-Spiele) und später von Chen und Deng (erweitert auf zwei-Spieler-Spiele) erbracht wurden.

## 6.4 Pareto Effizienz

Die Pareto Effizienz beschreibt eine Eigenschaft eines Zustands. Ein Zustand ist Pareto-effizient, wenn es keinen anderen Zustand gibt, durch den ein Spieler mehr Nutzen hat und gleichzeitig kein anderer Spieler weniger Nutzen.

Um zu verdeutlichen was damit gemeint ist, wird noch einmal das Prisoner's Dilemma herangezogen (siehe Tabelle 11). Wenn man die einzelnen Zustände auf die Eigenschaft Pareto-Effizienz prüft, wird klar, dass der Zustand  $(D, D)$ , also wenn beide Gefangenen die Aktion defect wählen, der einzige Zustand ist, der nicht Pareto-effizient ist. Zu diesem Zustand (für den das Nutzentupel  $(-2, -2)$  gilt) gibt es nämlich einen Zustand, bei dem alle Gefangenen einen größeren Nutzen erzielen könnten: Der Zustand  $(C, C)$  mit dem Nutzen  $(-1, -1)$  bringt beiden Gefangenen mehr Nutzen ein.

	$j$ defects	$j$ cooperates
$i$ defects	$(-2, -2)$	$(0, -3)$
$i$ cooperates	$(-3, 0)$	$(-1, -1)$

Tabelle 11: PRISONER'S DILEMMA GAME

## 6.5 Maximierung des Gemeinwohls

Die Grundlage des Prinzips der Maximierung des Gemeinwohls (social welfare) ist Summierung des Nutzens aller Agenten eines Zustands. Maximieren bedeutet in diesem Zusammenhang den Zustand suchen, der den größten summierten Nutzen hat. Die Anwendung dieses Prinzips ist sinnvoll, wenn es egal ist, welcher Agent mehr Nutzen hat und welcher weniger. Das ist der Fall, wenn alle Agenten den selben Besitzer haben.

## 7 Konstantsummenspiel

In einem Konstantsummenspiel kommt es zu einer Konkurrenzsituation. Diese wird strikte Konkurrenz genannt wenn die Verteilung der Nutzenanteile genau entgegengesetzt ist. Das bedeutet, dass ein Spieler genau die Summe weniger an Nutzen erhält, die der andere Spieler mehr an Nutzen bekommt. Beispiele für Konstante-Summen Spiele sind "Stein, Schere, Papier" (Payoff-Matrix abgebildet in Tabelle 9) und "matching pennies" (Payoff-Matrix abgebildet in Tabelle 10).

### 7.1 Nullsummenspiel

Ein Nullsummenspiel ist ein Konstantsummenspiel mit der Nutzensumme 0 für jeden Zustand. Beispiele dafür sind die unter den Konstantsummenspielen genannten Spiele.

## 8 The shadow of the future

Wenn man das Beispiel Prisoner's Dilemma um die Eigenschaft erweitert mehrere Runden nacheinander zu spielen und die Agenten (spielen die Gefangenen) sich die Aktionen aller Agenten aus den vorherigen Runden merken können, müssen die Agenten die Reaktion auf das eigene Verhalten einberechnen. Dieses Einberechnen der Reaktion wird als "shadow of the future" bezeichnet. Wenn man die Payoffs (siehe Tabelle 11) betrachtet ist die rationale Aktion für ein unendliche oft ausgeführtes Spiel cooperate, denn dadurch wird der Nutzen für beide Agenten zwar negativ pro Runde, aber nicht so negativ, wie wenn beide die Aktion defect wählen würden. Wird das Spiel endlich oft gespielt, gibt es eine Begründung für ein anderes Verhalten. Angenommen das Spiel wird 10 mal gespielt. Dann können die Agenten in der 10. Runde so spielen, als ob es nur ein Spiel mit einer Runde wäre, denn es gibt keine Konsequenz zu fürchten, weil es keine weitere Runde gibt. Ähnliche Erwägungen führen dazu, dass die vorletzte Runde wieder wie ein Spiel mit nur einer Runde gespielt werden kann. Das kann man bis zur ersten Runde fortführen. Diese Rückwärtsinduktion führt zu dem Schluss, dass bei bekannter Anzahl Runden, defect auch bei wiederholtem Spielen die rationale Aktion ist.

## 9 Programmgleichgewicht

Angenommen man lässt zwei Agenten das Prisoner's Dilemma (abgebildet in Tabelle 11) spielen. Zusätzlich gibt es noch einen Mittelsmann, der die Algorithmen auswerten kann. Dann können beide Agenten über diesen Mittelsmann indirekt miteinander verhandeln: Agent  $i$ 's und Agent  $j$ 's Algorithmen besagen z. B. wenn der jeweils andere Agent die Strategie cooperate wählt, wählt der Agent auch cooperate. Ansonsten kommen individuelle Strategien zum Einsatz.

Umsetzen lässt sich die Funktion des Mittelsmanns für dieses Beispiel im einfachsten Fall über einen Quelltextvergleich.

## 10 Umsetzung Multiagenten Szenarien

Damit ein Agent ein Spiel spielen kann, muss er die Regeln des Spiels und die Auswirkungen von Aktionen kennen. Nun gibt es zwei Möglichkeiten diese Informationen den Agenten verständlich zu machen:

1. Zur Entwurfszeit fest verdrahtet in den Agenten implementieren, oder
2. die Informationen in einem für die Agenten verarbeitbaren Format bereitstellen.

Die Nachteile der ersten Lösung liegen auf der Hand. Der Agent kann nicht dazulernen und kann nur Spiele spielen, die fest implementiert sind, was die Anzahl der Spiele begrenzt. Für die zweite Lösung hingegen gibt es eine von Stanford University zuletzt im Jahre 2008 geänderte praktische Umsetzung. Mit

Hilfe der Game Description Language (GDL) lassen sich endliche, diskrete und deterministische Mehrspielerspiele mit vollständiger Information für alle Spieler beschreiben. Computerprogramme bekommen im GDL-Format eine Anleitung, wie das Spiel funktioniert. Die Notation der GDL basiert auf Datalog, einem Ableger von Prolog. In dieser Notation werden die Zustände der Spielwelt durch eine Menge von wahren Aussagen ausgedrückt. Die Regeln des Spiels (Zustandsübergangsfunktionen) sind durch logische Regeln (Implikationen) umgesetzt, die die Menge der wahren Aussagen für den nächsten Zustand definieren. Welche Aktionen legal sind und welche Bedingungen das Spielende hat, wird auch durch logische Regeln festgelegt. Im Anhang befindet sich ein vollständiges Beispiel für das Spiel "Tic Tac Toe" (Drei gewinnt).

## Literatur

- [LHH<sup>+</sup>08] LOVE, Nathaniel ; HINRICHS, Timothy ; HALEY, David ; SCHKUFZA, Eric ; GENESERETH, Michael: *General Game Playing: Game Description Language Specification*. [http://games.stanford.edu/language/spec/gdl\\_spec\\_2008\\_03.pdf](http://games.stanford.edu/language/spec/gdl_spec_2008_03.pdf), 2008
- [OR94] OSBORNE, Martin J. ; RUBENSTEIN, Ariel: *A Course in Game Theory*. The MIT Press, 1994
- [SLB09] SHOHAM, Yoav ; LEYTON-BROWN, Kevin: *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009
- [Wik10a] WIKIPEDIA: *Nullsummenspiel*. 13.06.2010. <http://de.wikipedia.org/wiki/Nullsummenspiel>. Version: 2010
- [Wik10b] WIKIPEDIA: *Strategy (game theory)*. 13.06.2010. [http://en.wikipedia.org/wiki/Mixed\\_strategy](http://en.wikipedia.org/wiki/Mixed_strategy). Version: 2010
- [Woo09] WOOLDRIDGE, Michael: *An Introduction to Multiagent Systems*. John Wiley & Sons Ltd, 2009

## Teil I

# Anhang

### Vollständige Spezifikation des Spiels Tic Tac Toe in GDL (Game Description Language)

Das Semikolon kennzeichnet Kommentare. Das Symbol “<=” steht für eine Implikation.

Der nachfolgende Code-Ausschnitt bedeutet so: Wenn der Spieler am Zug ist (“control”) und die Zelle leer ist (“b“ für blank) lässt sich daraus folgern (“<=”): Es ist legal, dass diese Zelle für den Spieler markiert wird.

```
...
(<= (legal ?player (mark ?x ?y))
     (true (cell ?x ?y b))
     (true (control ?player)))
...
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Tic Tac Toe
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Rollen
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(role x)
(role o)
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Initialer Zustand
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(init (cell 1 1 b))
(init (cell 1 2 b))
(init (cell 1 3 b))
(init (cell 2 1 b))
(init (cell 2 2 b))
(init (cell 2 3 b))
(init (cell 3 1 b))
(init (cell 3 2 b))
(init (cell 3 3 b))
(init (control x))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Dynamische Komponenten
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Zelle
(<= (next (cell ?x ?y ?player))
```

```

        (does ?player (mark ?x ?y)))

(<= (next (cell ?x ?y ?mark))
    (true (cell ?x ?y ?mark))
    (does ?player (mark ?m ?n))
    (distinctCell ?x ?y ?m ?n))

;; Spielkontrolle wer am Zug ist
(<= (next (control x))
    (true (control o)))

(<= (next (control o))
    (true (control x)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Ansichten
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(<= (row ?x ?player)
    (true (cell ?x 1 ?player))
    (true (cell ?x 2 ?player))
    (true (cell ?x 3 ?player)))

(<= (column ?y ?player)
    (true (cell 1 ?y ?player))
    (true (cell 2 ?y ?player))
    (true (cell 3 ?y ?player)))

(<= (diagonal ?player)
    (true (cell 1 1 ?player))
    (true (cell 2 2 ?player))
    (true (cell 3 3 ?player)))

(<= (diagonal ?player)
    (true (cell 1 3 ?player))
    (true (cell 2 2 ?player))
    (true (cell 3 1 ?player)))

(<= (line ?player) (row ?x ?player))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Legals Zuege
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(<= (legal ?player (mark ?x ?y))
    (true (cell ?x ?y b))
    (true (control ?player)))

```

```

(<= (legal x noop)
     (true (control o)))

(<= (legal o noop)
     (true (control x)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Spielziele
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(<= (goal ?player 100)
     (line ?player))

(<= (goal ?player 50)
     (not (line x))
     (not (line o))
     (not open))

(<= (goal ?player1 0)
     (line ?player2)
     (distinct ?player1 ?player2))

(<= (goal ?player 0)
     (not (line x))
     (not (line o))
     open)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Endzustaende
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(<= terminal
     (line ?player))

(<= terminal
     (not open))

```