

Nebenläufige Erreichbarkeitsspiele

Zusammenfassung

Dieser Vortrag entstand im Rahmen des Hauptseminars „Spiele in der Informatik“ bei Prof. Javier Esparza im Sommersemester 2006 an der Universität Stuttgart.

Wir untersuchen nebenläufige Spiele, wie sie etwa bei der Modellierung offener Systeme auftreten. Dabei wählen Spieler 1 (das System) und Spieler 2 (die Umwelt) ihre Spielzüge gleichzeitig und unabhängig voneinander. Eigenschaften des Systems können als Ziele dieses Spiels modelliert werden.

Dabei beschränken wir uns auf deterministische Spiele mit endlichem Zustandsraum; trotz dieser Annahme gibt es einen deutlichen Unterschied zu zugbasierten Spielen: Man muss für Spieler 1 drei mögliche Gewinnsituationen unterscheiden, nämlich *sichere*, *fast sichere* und *im Grenzwert sichere* Gewinnzustände.

In diesem Vortrag werden wir eine formale Definition von nebenläufigen Erreichbarkeitsspielen geben und die jeweilige Bedeutung der drei Typen von Gewinnzuständen präzisieren. Anschließend werden wir sehen, dass sich alle drei Zustandsmengen in Polynomialzeit berechnen lassen, und entsprechende Gewinnstrategien angeben.

Grundlage ist dabei der im Literaturverzeichnis angegebene Artikel von L. de Alfaro, T. A. Henzinger und O. Kupferman.

17. Juli 2006

Mathias Häbich

1 Erreichbarkeitsspiele

Wir wollen den Begriff des Spiels in unserem Sinne formal fassen und und sagen:

Definition 1.1 Eine (*Zwei-Spieler-*)*Spielstruktur* ist ein Quintupel

$$G = (S, M, \Gamma_1, \Gamma_2, \delta),$$

bestehend aus:

- einem endlichen Zustandsraum S (den *Spielkonfigurationen*),
- einer endlichen Menge M von *Zügen* (*moves*),
- zwei Abbildungen $\Gamma_i : S \rightarrow 2^M \setminus \{\emptyset\}$, $i \in \{1, 2\}$, welche jedem Zustand die von diesem aus möglichen Züge von Spieler 1 bzw. 2 zuordnen,
- sowie einer (partiellen) Übergangsabbildung $\delta : S \times M \times M \rightarrow S$, welche jedem Tripel (s, m_1, m_2) mit $m_i \in \Gamma_i(s)$ einen Folgezustand zuordnet.

Aus technischen Gründen nehmen wir

$$\Gamma_i(s) = \Gamma_j(t) \implies i = j \wedge s = t$$

an, sodass alle Züge sowohl in allen Zuständen als auch für die beiden Spieler verschieden sind.

Weiterhin definieren wir:

Definition 1.2

- i) Ein *Pfad* in G ist eine Folge $\sigma = s_0 s_1 \dots \in S^\omega$, sodass für jedes $k \in \mathbb{N}_0$ je ein $m_i^k \in \Gamma_i(s_k)$, $i \in \{1, 2\}$, existiert mit $s_{k+1} = \delta(s_k, m_1^k, m_2^k)$. Die Menge aller Pfade bezeichnen wir mit Ω .
- ii) Eine Spielstruktur G heißt *zugbasiert*, falls für jeden Zustand $s \in S$ höchstens ein $i \in \{1, 2\}$ existiert mit $|\Gamma_i(s)| > 1$.

Definition 1.3 Ein *Erreichbarkeitsspiel* (oder kurz *Spiel*) ist ein Paar $\mathcal{G} = (G, R)$, bestehend aus einer Spielstruktur G und einer Menge $R \subseteq S$ von *Zielzuständen*.

Das Ziel von Spieler 1 ist es, einen Zustand in R zu erreichen, das Ziel von Spieler 2 ist es, dies zu verhindern. Der Einfachheit halber nehmen wir an, dass R *absorbierend* ist, d.h. $\delta(r, \Gamma_1(r), \Gamma_2(r)) \subseteq R$ für alle $r \in R$. Da Spieler 1 nur einen Zustand aus R *erreichen* muss, um zu gewinnen, ist es trivial, ein Spiel mit nicht-absorbierender Menge von Zielzuständen in ein äquivalentes Spiel mit absorbierender Zielmenge umzuwandeln.

Als *Größe* eines Spiels bezeichnen wir die Zahl der „Einträge“ der Übergangsabbildung δ , d.h.

$$|\mathcal{G}| = \sum_{s \in S} |\Gamma_1(s)| |\Gamma_2(s)|.$$

1.1 Zwei Beispiele

Die folgenden beiden Beispiele illustrieren nicht nur die Definitionen 1.2 und 1.3, sondern sind auch für die späteren theoretischen Überlegungen bedeutsam.

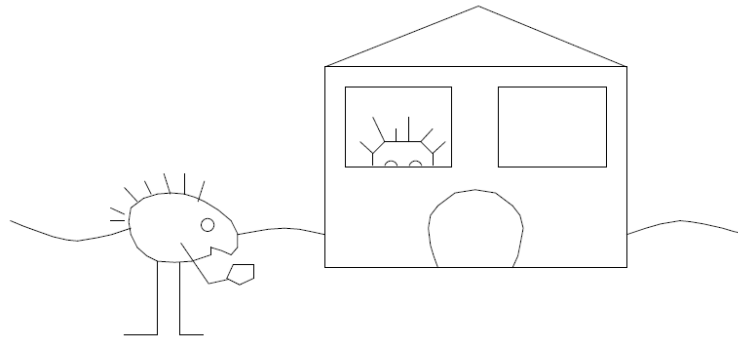


Abbildung 1: Das Spiel LINKS-ODER-RECHTS, Illustration aus [AHK].

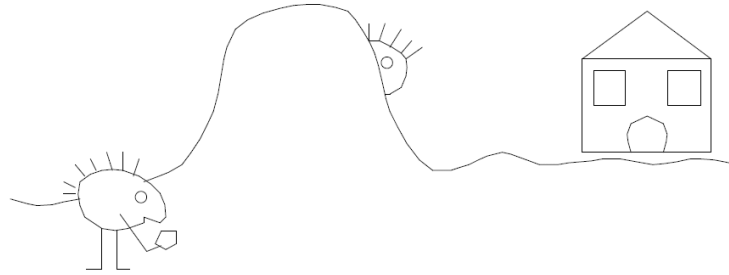
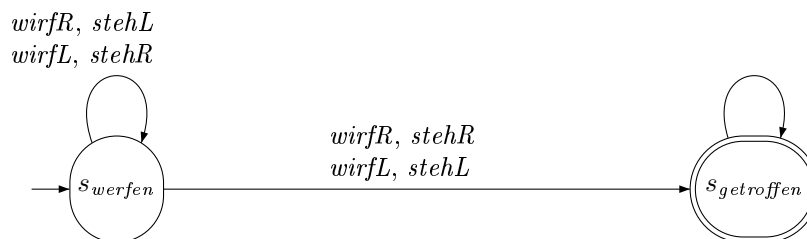


Abbildung 2: Das Spiel VERSTECKEN-ODER-RENNEN, Illustration aus [AHK].

1.1.1 Das Spiel LINKS-ODER-RECHTS

Spiel 1.4 (LINKS-ODER-RECHTS) Spieler 1 wirft in jeder Runde einen Schnellball entweder auf das linke oder auf das rechte Fenster eines Hauses. Gleichzeitig und unabhängig davon muss sich Spieler 2 entscheiden, entweder hinter dem linken oder hinter dem rechten Fenster zu stehen. Das Spiel endet mit einem Sieg für Spieler 1, wenn Spieler 2 von einem Schneeball getroffen wird.

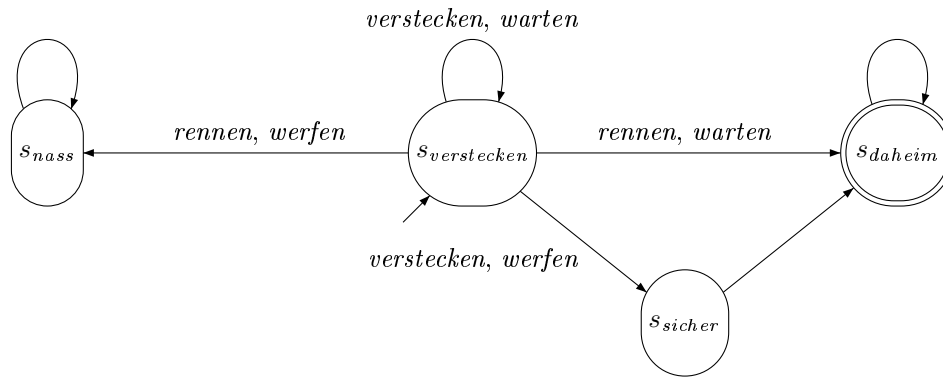
Das Spiel wird durch den folgenden Graphen beschrieben, wobei der Zielzustand durch einen doppelten Rahmen markiert ist:



1.1.2 Das Spiel VERSTECKEN-ODER-RENNEN

Spiel 1.5 (VERSTECKEN-ODER-RENNEN) Spieler 1 versteckt sich hinter einem Hügel und entscheidet in jeder Runde, ob er sich weiter verstecken oder nach Hause rennen will. Spieler 2 hat einen Schneeball und entscheidet in jeder Runde, ob er diesen werfen will oder nicht. Wirft Spieler 2, während Spieler 1 rennt, trifft er, und Spieler 2 gewinnt. Kann hingegen Spieler 1 nach Hause rennen, ohne dass Spieler 2 wirft, oder wirft Spieler 2 seinen (einzig) Schneeball, während sich Spieler 1 noch versteckt, gewinnt Spieler 1.

Das Spiel wird durch den folgenden Graphen beschrieben:



1.2 Strategien

Definition 1.6 (Strategie) Sei $\mathcal{D}(M)$ die Menge der Wahrscheinlichkeitsverteilungen auf der Menge der Züge einer Spielstruktur. Eine *Strategie für Spieler i* , $i \in \{1, 2\}$, ist eine Abbildung $\pi_i : S^+ \rightarrow \mathcal{D}(M)$, wobei für alle $\sigma \in S^*$ und $s \in S$ gelte, dass $\pi_i(\sigma s)(m) = 0$ für $m \notin \Gamma_i(s)$. (Eine Strategie schreibt also nur erlaubte Spielzüge vor.)

Die Menge aller Strategien für Spieler i bezeichnen wir mit Π_i .

1.2.1 Arten von Strategien

Wir unterscheiden verschiedene Arten von Strategien:

Definition 1.7 Eine Strategie π heißt

- i) *deterministisch*, falls für alle $\sigma \in S^+$ ein $m \in M$ existiert mit $\pi(\sigma)(m) = 1$,
- ii) *zählend*, falls $\pi(\sigma_1 s) = \pi(\sigma_2 s)$ für alle $\sigma_1, \sigma_2 \in S^*$ mit $|\sigma_1| = |\sigma_2|$, d.h. die Strategie hängt nur vom momentanen Zustand und von der Anzahl der gespielten Runden ab,
- iii) *von endlichem Gedächtnis*, falls es ein $n \in \mathbb{N}$ gibt mit $\pi(\sigma s_1 \cdots s_n) = \pi(s_1 \cdots s_n)$ für alle $\sigma \in S^*$ und $s_k \in S$, $k \in \{1, \dots, n\}$,
- iv) *gedächtnislos*, falls $\pi(\sigma s) = \pi(s)$ für alle $\sigma \in S^*$ und $s \in S$.

Die Diskussion von Gewinn- und Störstrategien erfolgt in Kapitel 1.3 („Gewinnzustände“), ihre Berechnung in Kapitel 2 („Berechnung der Gewinnzustände“).

1.3 Gewinnzustände

Gewinnzustände sind Spielzustände, aus denen Spieler 1 die Zielmenge R mehr oder weniger sicher erreichen kann; dabei müssen wir drei Arten von Gewinnzuständen unterscheiden:

Definition 1.8

- i) *Sichere Erreichbarkeit*: Mit $Sure(R)$ bezeichnen wir die Menge der Zustände, von denen aus Spieler 1 R *sicher* erreichen kann, d.h. $s \in Sure(R)$ genau dann, wenn es ein $\pi_1 \in \Pi_1$ gibt, sodass für alle $\pi_2 \in \Pi_2$ das Spiel irgendwann in R landet.
- ii) *Fast sichere Erreichbarkeit*: $Almost(R)$ sei die Menge der Zustände, von denen aus Spieler 1 R *fast sicher* erreichen kann. d.h. $s \in Almost(R)$ genau dann, wenn es $\pi_1 \in \Pi_1$ gibt, sodass für alle $\pi_2 \in \Pi_2$ das Spiel mit Wahrscheinlichkeit 1 in R landet.

- iii) *Grenzsichere Erreichbarkeit*: $Limit(R)$ sei die Menge der Zustände, aus denen R mit einer Wahrscheinlichkeit „beliebig nahe an 1“ erreicht werden kann, d.h. $s \in Limit(R)$ genau dann, wenn es für jedes $\varepsilon > 0$ eine Strategie $\pi_1 \in \Pi_1$ gibt, sodass für alle Strategien $\pi_2 \in \Pi_2$ das Spiel mit einer Wahrscheinlichkeit von mindestens $1 - \varepsilon$ schließlich R erreicht.

Offenbar gilt stets $Sure(R) \subseteq Almost(R) \subseteq Limit(R)$; es gibt Spiele, bei denen beide Inklusionen echt sind. Dies ist eine Besonderheit von nebenläufigen Spielen; bei zugbasierten fallen die drei Mengen zusammen!¹

Im Spiel LINKS-ODER-RECHTS ist die erste Inklusion echt, denn zu jeder deterministischen Strategie von Spieler 1 gibt es (genau eine) deterministische Strategie von Spieler 2, sodass der Zielzustand nie erreicht wird. Entscheidet Spieler 1 hingegen in jeder Runde durch den Wurf einer fairen Münze zufällig, auf welches Fenster er werfen will, so trifft er in jeder Runde mit Wahrscheinlichkeit $1/2$, also mit Wahrscheinlichkeit 1 irgendwann.

Im Spiel VERSTECKEN-ODER-RENNEN ist die zweite Inklusion echt, denn Spieler 1 kann das Haus nicht mit Wahrscheinlichkeit 1 erreichen: Entscheidet er sich, zu warten, bis Spieler 2 seinen Schneeball wirft, so wird er nie nachhause kommen, wenn Spieler 2 ebenfalls einfach abwartet. Beschließt er hingegen, zu einem Zeitpunkt mit Wahrscheinlichkeit $\varepsilon > 0$ zu rennen, so kann Spieler 2 zum ersten solchen Zeitpunkt werfen, sodass Spieler 1 mit Wahrscheinlichkeit ε getroffen wird. Eine Möglichkeit, mit beliebig hoher Wahrscheinlichkeit trocken nach Hause zu kommen, besteht darin, in jedem Zug mit konstant kleiner Wahrscheinlichkeit ε zu rennen.

1.3.1 Gewinn- und Störstrategien

Definition 1.9 Eine *Gewinnstrategie* für sichere (fast sichere, grenz-sichere) Erreichbarkeit ist eine Strategie² für Spieler 1, sodass das Spiel für jeden Zustand $s \in Sure(R)$ ($s \in Almost(R)$, $s \in Limit(R)$) für alle Strategien von Spieler 2 sicher (fast sicher, mit Wahrscheinlichkeit $1 - \varepsilon$) in R landet.

Eine *Störstrategie* für sichere (fast sichere, grenzsichere) Erreichbarkeit ist entsprechend eine Strategie für Spieler 2, sodass das Spiel für jeden Zustand $s \notin Sure(R)$ ($s \notin Almost(R)$, $s \notin Limit(R)$) für alle Strategien von Spieler 1 nicht sicher (mit Wahrscheinlichkeit < 1 , mit Wahrscheinlichkeit $< q < 1$ für festes q) in R landet.

Während sich die Existenz von Gewinnstrategien sofort aus der Definition der entsprechenden Gewinnzustände ergibt, ist die Existenz von Störstrategien nicht offensichtlich. Sie existieren jedoch, wie wir sehen werden, tatsächlich immer.

Im Folgenden werden wir Algorithmen vorstellen, die die verschiedenen Arten von Gewinnzuständen in Polynomialzeit berechnen.

2 Berechnung der Gewinnzustände

Definition 2.1 (Zugbeschränkung) Eine *Zugbeschränkung* für Spieler i ist eine Abbildung $\gamma_i : S \rightarrow 2^M$ mit $\gamma_i(s) \subseteq \Gamma_i(s)$. Die Menge der Zugbeschränkungen von Spieler 1 bezeichnen wir mit Δ .

Definition 2.2 Die Abbildung $Pre_1 : 2^S \times \Delta \rightarrow 2^S$ ist definiert über

$$Pre_1(U, \gamma) = \{s \in S \mid \exists m_1 \in \gamma(s) \forall m_2 \in \Gamma_2(s) : \delta(s, m_1, m_2) \in U\}.$$

Analog ist

$$Pre_2(U, \gamma) = \{s \in S \mid \exists m_2 \in \Gamma_2(s) \forall m_1 \in \gamma(s) : \delta(s, m_1, m_2) \in U\}.$$

Beachte, dass sich die Zugbeschränkung γ immer auf Spieler 1 bezieht!

Die Abbildungen Pre_i sind offenbar monoton im ersten Argument und verträglich mit Vereinigungen, d.h. es gilt $Pre_i(U \cup U', \gamma) = Pre_i(U, \gamma) \cup Pre_i(U', \gamma)$.

¹Das liegt daran, dass man sich bei deterministischen, endlichen, zugbasierten Spielen mit vollständiger Information auf deterministische Strategien für beide Spieler beschränken kann (es gibt immer einen „besten“ Zug). Eine Familie von Strategien für grenzsichere Erreichbarkeit enthält damit notwendig eine Strategie für fast sichere Erreichbarkeit, und fast sichere Erreichbarkeit wiederum ist äquivalent zu sicherer Erreichbarkeit.

²Im Falle grenzsicherer Erreichbarkeit natürlich eine Familie von Strategien!

2.0.2 Sichere Erreichbarkeit

Dies ermöglicht bereits die Berechnung von $Sure(R)$ vermöge einer Fixpunktiteration, wie der folgende Satz zeigt:

Satz 2.3 ($Sure(R)$)

- i) $Sure(R)$ ist der kleinste Fixpunkt der Abbildung

$$\Phi : 2^S \rightarrow 2^S, U \mapsto U \cup Pre_1(U, \Gamma_1),$$

welcher R enthält, und ist damit der Grenzwert der Folge

$$R \subseteq \Phi(R) \subseteq \Phi^2(R) \subseteq \dots$$

Dieser kann in linearer Zeit in der Größe der Spielstruktur bestimmt werden.

- ii) Spieler 1 besitzt eine deterministische, gedächtnislose Gewinnstrategie für sichere Erreichbarkeit.
 iii) Spieler 2 besitzt eine gedächtnislose Störstrategie für sichere Erreichbarkeit. Diese kann im Allgemeinen *nicht* deterministisch gewählt werden.
 iv) Die Zeit bis zum Erreichen von R ist für Zustände aus $Sure(R)$ beschränkt durch die Größe des Zustandsraumes.

Die Berechenbarkeit als Grenzwert folgt – wie in den folgenden Situationen ebenfalls – aus dem Fixpunktsatz von Knaster-Tarski; die Beweise für die anderen Aussagen und die sich ergebenden Gewinn- und Störstrategien unterscheiden sich nicht von denen in zugbasierten Spielen.

2.0.3 Fast sichere Erreichbarkeit

Die Berechnungen von $Almost(R)$ und $Limit(R)$ erfordern weitergehende Hilfsmittel:

Definition 2.4 Die Abbildungen $Safe_i : 2^S \times \Delta \rightarrow 2^S$ werden definiert über

$$Safe_i(U, \gamma) = \bigcup \{V \subseteq U \mid V \subseteq Pre_i(V, \gamma)\}.$$

$Safe_i$ ist monoton im ersten Argument.

Lemma 2.5

- i) Es gilt $Safe_i(U, \gamma) \subset U$ und $Safe_i(U, \gamma) \subseteq Pre_i(Safe_i(U, \gamma), \gamma)$. $Safe_i(U, \gamma)$ ist die größte Menge mit diesen Eigenschaften. Das bedeutet: $Safe_i(U, \gamma)$ ist die größte Teilmenge von U , für die Spieler i garantieren kann, sie im nächsten Zug nicht zu verlassen.
 ii) $Safe_i(U, \gamma)$ ist der größte Fixpunkt der Abbildung

$$\Psi : 2^S \rightarrow 2^S, V \mapsto V \cap Pre_i(V, \gamma),$$

welcher in U enthalten ist, und ist damit der Grenzwert der Folge

$$U \supseteq \Psi(U) \supseteq \Psi^2(U) \supseteq \dots$$

Dieser kann in linearer Zeit in der Größe der Spielstruktur bestimmt werden.

Definition 2.6 Die Abbildung $Stay_1 : 2^S \rightarrow \Delta$ wird definiert über

$$Stay_1(U)(s) = \{m_1 \in \Gamma_1(s) \mid \forall m_2 \in \Gamma_2(s) : \delta(s, m_1, m_2) \in U\}.$$

Algorithmus 1 $Almost(R)$

Eingabe: Erreichbarkeitsspiel $\mathcal{G} = (G, R)$

Ausgabe: $Almost(R)$

function ALMOST(\mathcal{G})

$U_0 = S$

$\gamma_0 = \Gamma_1$

for $k \geq 0$ **do**

$C_k = Safe_2(U_k \setminus R, \gamma_k)$

$U_{k+1} = Safe_1(U_k \setminus C_k, \gamma_k)$

$\gamma_{k+1} = Stay_1(U_{k+1})$

if $U_{k+1} = U_k$ **then**

return U_k

end if

end for

end function

$Stay_1(U)$ ist die größte Zugbeschränkung für Spieler 1, die sicherstellt, dass das Spiel nach einer Runde in U liegt.

Algorithmus 1 berechnet $Almost(R)$. Die Terminierung ist hierbei durch $U_{k+1} \subseteq U_k$ gesichert. Die Korrektheit folgt aus der Existenz von Gewinn- bzw. Störstrategien für Spieler 1 und 2 respektive, dazu sei im folgenden m minimal mit $U_{m+1} = U_m$.

Gewinnstrategie für Spieler 1: Für $s \in U_m$ ziehe zufällig gemäß $\gamma_m(s)$.

Da das Spiel damit U_m nicht verlässt und Spieler 2 das Spiel nicht auf $U_m \setminus R$ beschränken kann (Es ist $C_m = \emptyset!$), erreicht das Spiel mit Wahrscheinlichkeit 1 irgendwann einen Zustand in R .

Störstrategie für Spieler 2: Für $s \in C_k$ ziehe in Runde ℓ mit Wahrscheinlichkeit $q_\ell = 2^{-1/2^\ell}$ zufällig gemäß $Stay_2(C_k, \gamma_k)(s)$, oder mit Wahrscheinlichkeit $1 - q_\ell$ zufällig gemäß $\Gamma_2(s)$. Für $s \notin \bigcup_k C_k$, $s \notin U$ ziehe zufällig gemäß $\Gamma_2(s)$.

Diese etwas technische Vorschrift garantiert, dass das Spiel mit positiver Wahrscheinlichkeit in C_k „gefangen“ bleibt, wenn es dort startet und Spieler 1 gemäß γ_k zieht. Zieht Spieler 1 nicht gemäß γ_k , so fällt es mit positiver Wahrscheinlichkeit in $U_{k-1} \setminus U_k$, wo es – sofern es in C_{k-1} landet – wieder mit positiver Wahrscheinlichkeit gefangen wird usw. Insgesamt kann Spieler 1 nicht fast sicher gewinnen.

Zusammenfassend gilt:

Satz 2.7

- i) Algorithmus 1 berechnet $Almost(R)$ in quadratischer Zeit.
- ii) Spieler 1 besitzt eine gedächtnislose randomisierte Gewinnstrategie für fast sichere Erreichbarkeit. Diese kann im Allgemeinen nicht deterministisch gewählt werden.
- iii) Spieler 2 besitzt eine zählende randomisierte Störstrategie für fast sichere Erreichbarkeit. Diese kann im Allgemeinen nicht als von endlichem Gedächtnis gewählt werden.
- iv) Die Zeit bis zum Erreichen von R ist unbeschränkt. (Die *erwartete* Zeit ist jedoch beschränkt.)

Punkt iv) folgt aus stochastischen Überlegungen. Am Beispiel des Spiels LINKS-ODER-RECHTS beobachtet man, dass Spieler 1 mit der berechneten Gewinnstrategie („wähle das Fenster durch den Wurf einer fairen Münze“) in jeder Runde mit Wahrscheinlichkeit $1/2$ gewinnt. Die erwartete Spieldauer ist damit $\sum_{k=0}^{\infty} k \cdot 2^{-k} = 2$. Dass die Gewinnstrategie von Spieler 1 nicht immer deterministisch sein kann, sieht man am Spiel LINKS-ODER-RECHTS, das Spieler 1 offenbar fast sicher gewinnen kann, jedoch nicht mit einer deterministischen Strategie. (Spieler 2

Algorithmus 2 Grenzflüchtigkeit

Eingabe: Spielstruktur G , Zustandsmengen $C \subseteq U \subseteq S$, Zustand $s \in C$

Ausgabe: JA, falls s grenzflüchtig ist bezüglich C und U , NEIN sonst

function LIMIT(G, C, U, s)

$\mathcal{B}_{-1} = \emptyset$

repeat for $k \geq 0$

$\mathcal{A}_k = \{a \in \Gamma_1(s) : \forall b \in \Gamma_2(s) (\delta(s, a, b) \notin U \Rightarrow b \in \mathcal{B}_{k-1})\}$

$\mathcal{B}_k = \{b \in \Gamma_2(s) : \exists a \in \mathcal{A}_k \delta(s, a, b) \notin C\}$

until $\mathcal{A}_{k+1} = \mathcal{A}_k \wedge \mathcal{B}_{k+1} = \mathcal{B}_k$

if $\mathcal{B}_k = \Gamma_2(s)$ **then**

return JA

else

return NEIN

end if

end function

besitzt dann genau eine deterministische Gegenstrategie.) Dass die Störstrategie von Spieler 2 nicht immer von endlichem Gedächtnis sein kann, macht man sich anhand des Spiels VERSTECKEN-ODER-RENNEN klar: Wirft Spieler 2 irgendwann mit Wahrscheinlichkeit $\varepsilon > 0$, so tut er dies immer wieder, und Spieler 1 braucht nur abzuwarten. Anderenfalls kann Spieler 1 in der ersten Runde rennen.

2.0.4 Grenzsichere Erreichbarkeit

Auch $Limit(R)$ wird als Grenzwert einer absteigende Folge $S = U_0, U_1, \dots$ berechnet. Der Algorithmus bestimmt dazu in jeder Iteration eine Menge $C_k \subset U_k \setminus R$ von Zuständen, die nicht in $Limit(R)$ liegen, und setzt $U_{k+1} = U_k \setminus C_k$.

C_k wird ebenfalls iterativ bestimmt. Dazu setzen wir $C_k^0 = U_k \setminus R$ und erhalten durch Entfernen von Zuständen wieder eine absteigende Folge C_k^0, C_k^1, \dots . Betrachte dazu einen Zustand $s \in C_k^j$. Nach Konstruktion sind die Zustände $S \setminus U_k$ keine Gewinnzustände, also muss Spieler 1 verhindern, U_k zu verlassen. Gleichzeitig muss er wegen $C_k^j \cap R = \emptyset$ versuchen, C_k^j irgendwann zu verlassen.

Seien $\xi_1 \in \mathcal{D}(\Gamma_1(s))$, $\xi_2 \in \mathcal{D}(\Gamma_2(s))$ die von Spieler 1 und 2 verwendeten Verteilungen. Bezeichnet $p(s, \xi_1, \xi_2)(V)$ für eine Menge $V \subseteq S$ die Wahrscheinlichkeit, dass das Spiel von s aus in einem Zug einen Knoten in V erreicht, so betrachten wir das Verhältnis

$$\frac{p(s, \xi_1, \xi_2)(S \setminus C_k^j)}{p(s, \xi_1, \xi_2)(S \setminus U_k)}. \quad (*)$$

Kann Spieler 1 dieses Verhältnis durch geeignete Wahl von ξ_1 beliebig groß machen, so kann Spieler 2 s nicht dazu verwenden, das Spiel auf C_k^j zu beschränken. Denn wird s beliebig oft besucht, so wird die Wahrscheinlichkeit, dass das Spiel C_k^j verlässt, beliebig groß, während die Gefahr für Spieler 1, das Spiel durch das Verlassen von U_k zu verlieren, beliebig gering ist.

Kann Spieler 1 das Verhältnis (*) jedoch nicht beliebig groß machen, so steht der Wahrscheinlichkeit, C_k^j zu verlassen, eine proportionale Gefahr gegenüber, U_k zu verlassen. Spieler 1 kann s also nicht nutzen, um C_k^j zu verlassen, ohne Gefahr zu laufen, das Spiel zu verlieren. Dies motiviert die folgende

Definition 2.8 (Grenz-Flucht-Zustände) Für $U, C \subseteq S$ heißt $s \in S$ *Grenz-Flucht-Zustand* (*limit-escape state*) bezüglich C und U , wenn

$$\sup_{\xi_1 \in \mathcal{D}(\Gamma_1(s))} \inf_{\xi_2 \in \mathcal{D}(\Gamma_2(s))} \frac{p(s, \xi_1, \xi_2)(S \setminus C)}{p(s, \xi_1, \xi_2)(S \setminus U)} = \infty.$$

Algorithmus 3 $Cage(U)$

Eingabe: Erreichbarkeitsspiel $\mathcal{G} = (\mathbf{G}, R)$, Zustandsmenge $U \subseteq S$ mit $R \subseteq U$

Ausgabe: $Cage(U)$

function $CAGE(\mathcal{G}, U)$

$C^0 = U \setminus R$

repeat for $j \geq 0$

$C^{j+1} = \{s \in C^j : s \text{ nicht grenzflüchtig bezüglich } C^j \text{ und } U\}$

until $C^{j+1} = C^j$

return C^j

end function

Algorithmus 4 $Limit(R)$

Eingabe: Erreichbarkeitsspiel $\mathcal{G} = (\mathbf{G}, R)$

Ausgabe: Grenzsichere Erreichbarkeitszustände $Limit(U)$

function $LIMIT(\mathcal{G})$

$U_0 = S$

repeat for $k \geq 0$

$U_{k+1} = U_k \setminus Cage(U_k)$

until $U_{k+1} = U_k$

return U_k

end function

Der Algorithmus entfernt nun gerade die bezüglich C_k^j und U_k grenzflüchtigen Zustände aus C_k^j , um C_k^{j+1} zu bilden; diese berechnet Algorithmus 2.

Die Korrektheit sieht man wie folgt: Wir nennen einen Spielzug $a \in \Gamma_1(s)$ *markiert*, falls es ein k gibt mit $a \in \mathcal{A}_k$, und bezeichnen das kleinste solche k mit $\ell(a)$. Analog verfahren wir für Spielzüge $b \in \Gamma_2(s)$ mit \mathcal{B}_k statt \mathcal{A}_k .

Angenommen, der Algorithmus liefert JA als Ergebnis. Dann sind alle Züge von Spieler 2 markiert, und damit auch alle Züge von Spieler 1. Dies liefert eine ε -indizierte Familie von Verteilungen, die (*) divergieren lässt: Sei für $\varepsilon > 0$ genügend klein $\xi_1[\varepsilon]$ diejenige Verteilung auf $\Gamma_1(s)$, die einen Spielzug a mit $\ell(a) > 0$ mit Wahrscheinlichkeit $\varepsilon^{\ell(a)}$ spielt, und die übrigen Spielzüge mit $\ell(a) = 0$ rein zufällig mit der verbliebenen Wahrscheinlichkeit.

Um zu zeigen, dass (*) für $\varepsilon \rightarrow 0$ divergiert, betrachte einen Spielzug $b \in \Gamma_2(s)$. Dieser ist markiert, und es gibt einen Spielzug $a \in \Gamma_1(s)$ mit $\ell(a) = \ell(b)$. Die Wahrscheinlichkeit, beim Spielen von b in dieser Runde C zu verlassen, ist damit mindestens proportional zur Wahrscheinlichkeit, mit der a gespielt wird, oder $\varepsilon^{\ell(b)}$. Andererseits besitzt jeder Zug a , der, zusammen mit b gespielt, zum Verlassen von U führt, eine echt größere Markierung als b , sodass die Wahrscheinlichkeit, in diesem Zug U zu verlassen, höchstens proportional zu $|\Gamma_1(s)|\varepsilon^{\ell(b)+1}$ ist. Dieses Argument funktioniert für alle Züge von Spieler 2, sodass das Verhältnis (*) für $\varepsilon \rightarrow 0$ divergiert und s also grenzflüchtig ist.

Angenommen, der Algorithmus antwortet NEIN. Dann sind einige Züge von Spieler 2 nicht markiert; wir nehmen an, Spieler 2 spiele diese rein zufällig. Für einen Zug a von Spieler 1 gibt es nun zwei Möglichkeiten:

- a ist markiert. Dann bleibt das Spiel beim Spielen von a und einem beliebigen unmarkierten Zug b in C gefangen, (*) divergiert also nicht.
- a ist nicht markiert. Dann gibt es einen nichtmarkierten Zug b , der, zusammen mit a gespielt, zum Verlassen von U führt. Da dieser mit konstanter Wahrscheinlichkeit gespielt wird, ist die Wahrscheinlichkeit, U zu verlassen, proportional zu der Wahrscheinlichkeit, dass a gespielt wird, während die Wahrscheinlichkeit, C zu verlassen, entweder 0 ist oder ebenfalls proportional zur Wahrscheinlichkeit, dass a gespielt wird. Also kann (*) nicht divergieren.

Damit haben wir alle Bausteine zusammen, um die grenzsicheren Gewinnzustände zu berechnen. Wir geben den Algorithmus in Pseudocode sowie den entsprechenden Satz aus Gründen der Vollständigkeit an.

Algorithmus 3 berechnet zu einer Zustandsmenge U sowie einer Zielmenge R die größte Teilmenge $C \subset U \setminus R$, die keine grenzflüchtigen Zustände bezüglich C und U enthält; diese nennen wir $Cage(U)$. Algorithmus 4 berechnet schließlich $Limit(R)$.

Satz 2.9

- i) Algorithmus 4 berechnet $Limit(R)$ (bei geeigneter Implementierung) in quadratischer Zeit.
- ii) Spieler 1 besitzt eine Familie gedächtnisloser Gewinnstrategien für grenzsichere Erreichbarkeit; diese kann im Allgemeinen nicht deterministisch gewählt werden.
- iii) Spieler 2 besitzt eine gedächtnislose Störstrategie. Diese kann im Allgemeinen nicht deterministisch gewählt werden.

Um den Algorithmus wirklich mit quadratischer Laufzeit zu implementieren, muss Algorithmus 3 optimiert werden; siehe dazu [AHK].

Sei m minimal mit $U_{m+1} = U_m$. Dann lauten die Strategien für die beiden Spieler wie folgt:

Gewinnstrategie für Spieler 1: Für $s \in C_m^j \setminus C_m^{j+1}$ ziehe gemäß der Strategie, die zeigt, dass s grenzflüchtig ist bzgl. C_m^j und U_m .

Störstrategie für Spieler 2: Für $s \in U_k \setminus U_{k+1} \subseteq Cage(U_k)$ ziehe gemäß der Strategie, die zeigt, dass s nicht grenzflüchtig ist bzgl. $Cage(U_k)$ und U_k .

Ein Beispiel für die Nichtexistenz gedächtnisloser Gewinnstrategien ist wieder das Spiel LINKS-ODER-RECHTS; dass die Störstrategie von Spieler 2 nicht immer deterministisch gewählt werden kann, sieht man an der Variante des Spiels, die nach einer Runde mit einem Sieg für Spieler 2 endet, wenn er nicht getroffen wurde.

Literatur

[AHK] L. de Alfaro, T. A. Henzinger, O. Kupferman. *Concurrent Reachability Games*. EECS Department, University of California, Berkeley. 1998.