Ms. Pac-Man gespielt durch Agenten mit einfachen Regeln

TECHNISCHE UNIVERSITÄT DARMSTADT

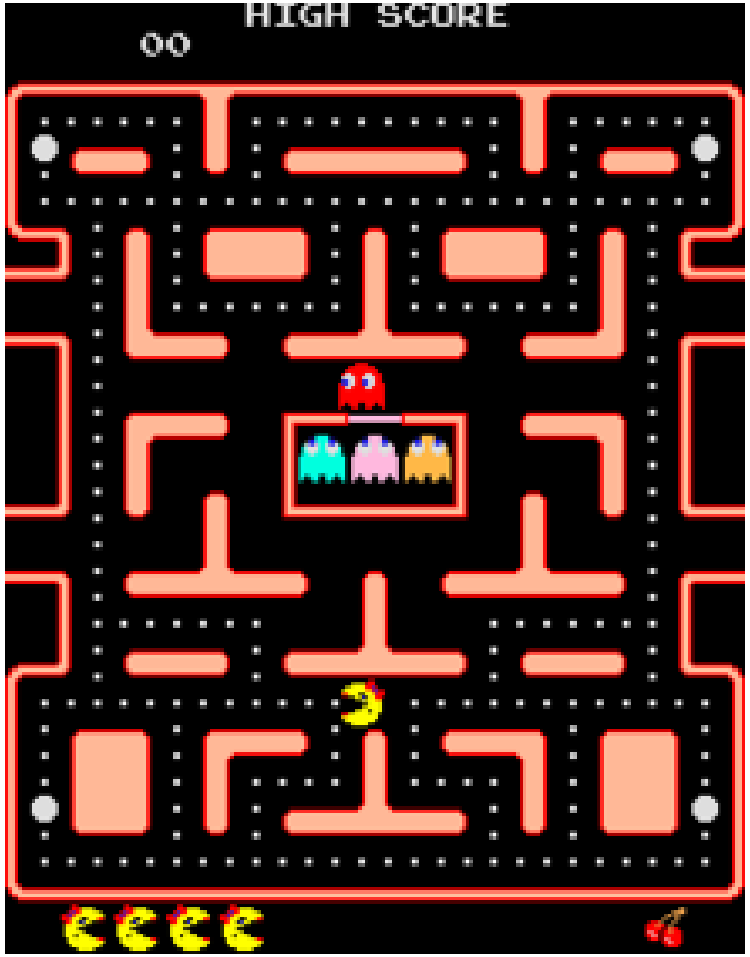
Seminarvortrag von Michael Schmitt

Übersicht

TECHNISCHE UNIVERSITÄT DARMSTADT

- Spielbeschreibung
- Reinforcement Learning
- Agent
  - Beobachtungen
  - Handlungen
  - Regeln
- Demo
- Lernmethoden
  - Statistical Gradient Method
  - Cross Entropy Method
  - Lernbasis
- Ergebnisse
- Warum „einfache“ Regeln?

Spielbeschreibung



- Arcarde Spiel (1981)
- Ziele:
  - Geister vermeiden
  - „Punkte“ fressen (10)
  - „Kraft-Punkte“ fressen (40)
  - Verwundbare Geister fressen
    - Für konsekutiv gefressene Geister:(200, 400, 800, 1600)
- Vier Leben, dann Game Over

Warum Ms. Pac-Man?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Spiele gut für Reinforcement Learning
  - Feste Regeln
    - Kein Anpassen der Regeln der Aufgabe an das Verfahren
  - Spiele sollen den menschlichen Intellekt fordernde Aufgaben stellen
  - Folge von Entscheidungen die teilweise weitreichend sind

- Original Pac-Man ungeeignet
  - Wege der Geister nach festem Muster
    - Bei Ms. Pac-Man nicht, sondern Verfolgung und Zufall
    - Stochastische Umgebung zwingend für Reinforcement Learning
    - Für Pac-Man selbst gibt es optimale statische Bewegungs Reihenfolge

Reinforcement Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zustandsmenge
- Handlungsmenge
- Belohnungswerte

- Zu jedem Zeitpunkt
  - Agent erhält aktuellen Zustand und mögliche Handlungen
  - Wählt eine Handlung
  - Daraus ergibt sich Folgezustand und Belohnung

- Gesucht
  - Strategie für Agenten die Belohnungssumme maximiert

- Iterativer Prozess
  - Lernen durch Bestätigung von erfolgreichem Handeln

Zustandsmenge und Handlungsmenge

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zustand:
  - Jeder „Punkt“ vorhanden oder nicht
  - Position der Geister
  - Position von Ms. Pac-Man
  - Ist „Kraft-Punkt“ aktiv und wenn ja, wie lange
- Handlung:
  - Wahl einer der möglichen Richtungen (höchstens 4) pro Zug
- Beschreibung des Zustands zu groß
- Handlungen haben keinen tieferen Sinn

  → Es ist schwer, „einfache“ Regeln zu formulieren

Abstrakte Beobachtungen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Merkmale des Zustands
- Unter Zuhilfenahme von Domainwissen formuliert
  - Ermöglichen die Formulierung von Strategien
- Möglichst einfach → geringer Programmieraufwand

- Beispiele:
  - Abstand zum nächsten Geist
  - Maximale „Sicherheit" der nächsten Kreuzungen
  - Abstand zum nächsten Punkt
  - Abstand zum nächsten Kraft-Punkt

Abstrakte Handlungen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Handlungen abhängig vom Kontext
  - Mehr als nur „links, rechts, oben, unten“
  - Gleiche Vorsätze wie bei Beobachtungen

- Beispiele:
  - Vor dem nächsten Geist fliehen
  - Zum nächsten Punkt gehen
  - Zum nächsten Kraft-Punkt gehen
  - Zur Mitte der Punkte gehen

- Aktive Handlungen auf drei Prioritätsebenen
  - Jede Handlung bewertet alle Richtungen
  - Jede Ebene reicht nur die besten Richtungen weiter
  - Agent wählt zufällig aus übrigen Richtungen

Agent - Strategie

TECHNISCHE UNIVERSITÄT DARMSTADT

- Zwei Instanzen des Agenten unterscheiden sich nur in ihrer Strategie
- Strategie ist zusammengesetzt aus mehreren Regeln
  - Regeln haben definierte Reihenfolge
- Bei jedem Schritt von Ms. Pac-Man wird Strategie einmal befragt
  - Erste Regel deren Prämisse zutrifft wird angewandt

Regeln

TECHNISCHE UNIVERSITÄT DARMSTADT

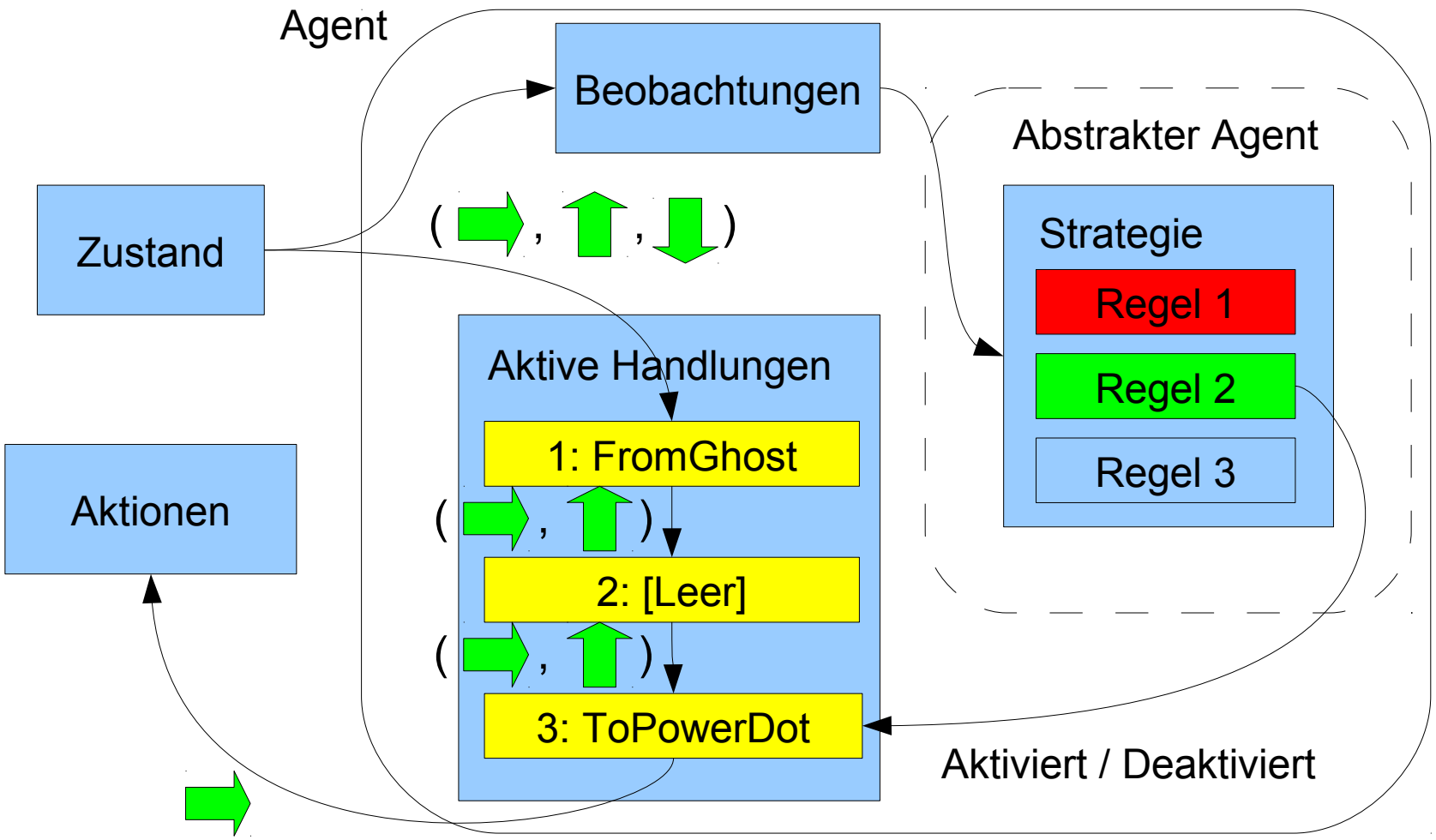
- Regeln haben die Form

  If [Beobachtung] [>|<] [Zahl] then [Handlung][+|-]

  - Wenn eine Regel angewandt wird, wird die entsprechende Handlung [+]aktiviert bzw. [-]deaktiviert

- Haben eine Priorität aus [1,2,3]
  - Aktivierte Handlung hat gleiche Priorität wie Regel
  - Aktivierte Handlung verdrängt andere Handlung mit gleicher Priorität
  - Deaktivierung unabhängig von Priorität

Entscheidungsfindung Agent



Demo

TECHNISCHE UNIVERSITÄT DARMSTADT

Implementierung der Demo

TECHNISCHE UNIVERSITÄT DARMSTADT

- Entscheidungsfindung des Agenten
  - Alle im Artikel beschriebenen Beobachtungen
  - Einige Aktionen
  - Möglichkeit zur Implementierung beliebiger Strategien
  - Gezeigt: Beste Strategie
- Keine Lernansätze implementiert
  - Suche nach optimaler Strategie fehlt
  - Benutzung einer externen Machine Learning Toolbox wie etwa shogun-python möglich

Reinforcement Learning Fortsetzung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Direkte Suche nach optimaler Strategie
- Mittels Verfahren aus der Optimierung
  - Statistical Gradient Method
  - Cross Entropy Method

- Ein Agent spielt mit einer Strategie mehrere Spiele
- Durchschnittliche Punktzahl wird verwendet

Statistical Gradient Method

TECHNISCHE UNIVERSITÄT DARMSTADT

- Strategie wird zufällig variiert (100 mal)
  - Pro Regel
    - Änderung der Prämisse mit p=5%
    - Änderung der Handlung mit p=5%
- Beste Variation wird ausgewählt
- Diese Strategie wird als nächstes variiert

Cross Entropy Method

TECHNISCHE UNIVERSITÄT DARMSTADT

- 1000 Zufällige Strategien werden erstellt
  - Die besten 50 werden ausgewählt
  - Deren Verteilung wird analysiert
  - Die nächsten Strategien werden dieser Verteilung entsprechend gewählt
- Verfahren ähnlich der Statistical Gradient Method
  - Nicht eine einzelne Strategie wird gewählt
  - Ausreißer bei der Bewertung nicht so entscheidend

Basis des Lernens

TECHNISCHE UNIVERSITÄT DARMSTADT

- Manuell erzeugte Regel Basis
  - Regeln die sinnvoll erscheinen werden erstellt
  - Gelernt wird
    - Welche verwendet werden
    - In welcher Reihenfolge
    - Mit welcher Priorität
- Zufällig erzeugte Regeln
  - Zufällig ausgewählte:
    - Merkmale
    - Schranken
    - Handlungen

Ergebnisse

TECHNISCHE UNIVERSITÄT DARMSTADT

Method	Avg. Score	(25%/75% percentiles)
CE-RANDOMRB	6382	(6147/6451)
CE-FIXEDRB	**8186**	(6682/9369)
SG-RANDOMRB	4135	(3356/5233)
SG-FIXEDRB	5449	(4843/6090)
CE-RANDOMRB-1ACTION	5417	(5319/5914)
CE-FIXEDRB-1ACTION	5631	(5705/5982)[6]
SG-RANDOMRB-1ACTION	2267	(1770/2694)
SG-FIXEDRB-1ACTION	4415	(3835/5364)
Random policy	676	(140/940)
Hand-coded policy	7547	(6190/9045)
Human play	8064	(5700/10665)

Warum „einfache“ Regeln?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Einfache Regeln werden durch Abstraktionsebene möglich
- Einfache Regeln werden erreicht durch ein Ungleichgewicht bei den Machine Learning Verfahren

- Von Menschen verfolgte Strategien haben oft geringe Komplexität
  - Es existieren also Lösungen mit geringer Komplexität

- Suchraum wird eingeschränkt
  - Es wird keine Zeit bei der Suche im großen Raum vergeudet

Anmerkungen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Einige abstrakte Handlungen sind nicht beschrieben, tauchen nur auf
- Asymmetrie zwischen Handlungen und Beobachtungen
  - Bsp.: TSP-Distanz hat kein äquivalent auf der Handlungsseite, wird folglich wohl nie sinnvoll zum Einsatz kommen
- Nur eine Regel wird aktiviert
  - Regeln nach Tautologien werden ignoriert
  - Dieser Fall kommt sogar im erklärenden Beispiel zweimal vor, wird aber ignoriert

Quellen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Istvan Szita, Andras Lörincz: Learning to Play Using Low-Complexity Rule-Based Policies: Illustrations through Ms. Pac-Man, in: Journal of Articial Intelligence Research 30 (2007) 659-684
- Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning, in: Machine Learning, 8, 229-256 (1992)
- A Tutorial on the Cross-Entropy Method. Annals of Operations Research. Vol. 134(1), pp 19-67, 2005

- Bilder:
  - http://goregirl.files.wordpress.com/2010/01/ms-pacman.jpg
  - http://en.wikipedia.org/wiki/Ms._Pac-Man
- Implementierung:
  - http://www.pygame.org/

Danke

TECHNISCHE UNIVERSITÄT DARMSTADT

- Vielen Dank für Ihre / Eure Aufmerksamkeit

Diskussion - Entscheidungsfindung

TECHNISCHE UNIVERSITÄT DARMSTADT

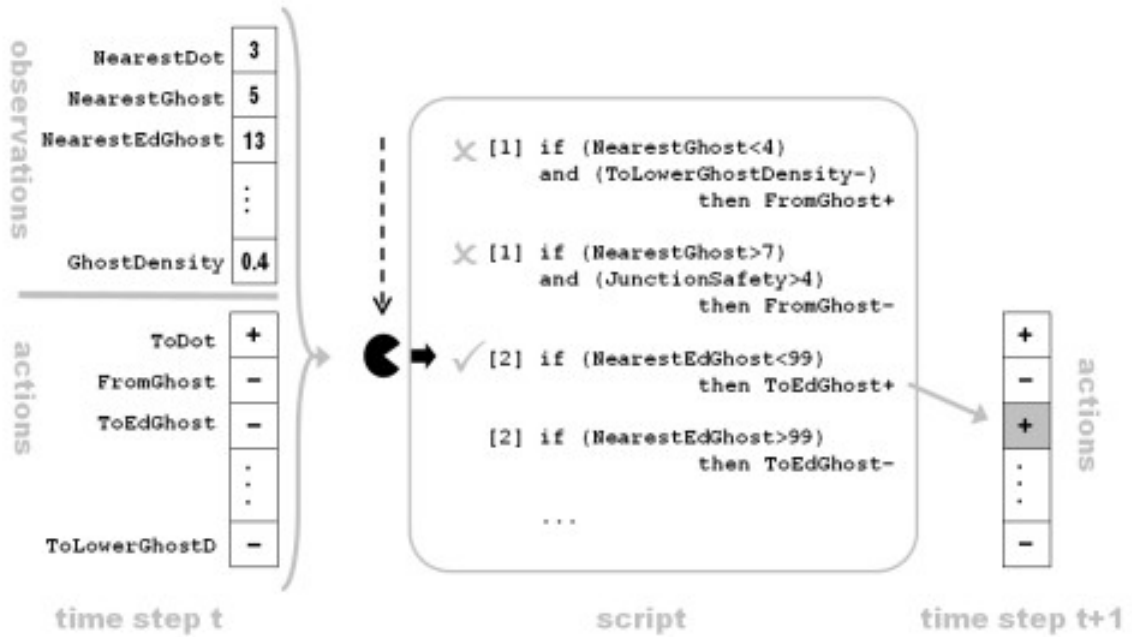


Figure 2: **Decision-making mechanism of the Ms. Pac-Man agent.** At time step t, the agent receives the actual observations and the state of her action modules. She checks the rules of her script in order, and executes the first rule with satisfied conditions.

Diskussion – Strategie mit Tautologie

TECHNISCHE UNIVERSITÄT DARMSTADT

Table 5: **A hand-coded policy for playing Ms. Pac-Man.** Bracketed numbers denote priorities, [1] is the highest priority.

```
[1] if NearestGhost<4 then FromGhost+
[1] if NearestGhost>7 and JunctionSafety>4 then FromGhost-
[2] if NearestEdGhost>99 then ToEdGhost-
[2] if NearestEdGhost<99 then ToEdGhost+
[3] if Constant>0 then KeepDirection+
[3] if FromPowerDot- then ToPowerDot+
[3] if GhostDensity<1.5 and NearestPowerDot<5 then FromPowerDot+
[3] if NearestPowerDot>10 then FromPowerDot-
```

Figure 3: Best policy learned by CE-FIXEDRB. Average score over 50 games: 9480.

```
[1] if MaxJunctionSafety>2.5 and ToLowerGhostDensity- then FromGhost-
[1] if NearestGhost<6 and MaxJunctionSafety<1 then FromGhost+
[1] if NearestGhost>6 and FromGhostCenter- then ToEdGhost+
[2] if ToEdGhost- and CenterOfDots>20 then ToEdGhost+
[2] if ToEdGhost- and NearestEdGhost<99 then ToEdGhost+
[2] if NearestDot>1 and GhostCenterDist>0 then KeepDirection+
[3] if ToGhostFreeArea- and ToDot- then ToPowerDot+
```