PRÜFUNGSVORBEREITUNG
SOMMERSEMESTER 2013

CHEATSHEET

FOUNDATIONS OF ARTIFICIAL INTELLIGENCE

6. September 2013

N.R.
BSc Informatik, 4. Fachsemester

Sommersemester 2013

Inhaltsverzeichnis

1 Suchen und Planung

1.1 Depth-Search

Einfachste Form der Suche. Findet Lösung, indem Baum in kompletter Tiefe besucht wird, bevor abzweigende Pfade aufgegriffen werden. Kann durch Variationen, wie die *beschränkte Tiefensuche*, oder daraus folgend die *iterative Tiefensuche*, verbessert werden.

1.2 A*

Informierter Suchalgorithmus, der den kürzesten Pfad zwischen Knoten innerhalb eines Graphen mit positiven Kantengewichten finden will. Verwendet eine Schätzfunktion (Heuristik), um zielgerichtet zu agieren.
Findet immer die optimale Lösung.

Algorithmus Wendet alle potentiellen Aktionen an und berechnet für jeden resultierenden Knoten einen Wert $f(x)$, der angibt, wie lange der Pfad von Start nach Ziel unter Verwendung des betrachteten Knotens x im günstigsten Fall ist. Anschließend wird die Aktion ausgeführt, die $f(x)$ minimiert:

$$f(x) = g(x) + h(x)$$

$g(x)$ bezeichnet dabei die Knoten vom Start aus zu x. $h(x)$ steht für die geschätzten Kosten zum Ziel (Heuristik).

1.3 Heuristiken

Heuristiken sind Schätzfunktionen, die Alternativen in Suchproblemen bewerten, um die lohnenswerteste Aktion zu identifizieren.

Heuristiken sind dabei so konstruiert, dass sie schnell ausführbar sind und ausschließlich als Schätzwerte dienen. Tatsächliche Informationen über das konkrete Problem und die damit verbundenen Kosten können nicht extrahiert werden.

1.3.1 Optimistische Heuristiken (admissible)

Optimistische Heuristiken sind so definiert, dass sie niemals die tatsächlichen Kosten zur Zielerreichung überschätzen. Der Schätzwert ist also niemals höher als die geringstmöglichen Kosten.

1.4 Constraint-Satisfaction-Problem

CSP's sind Probleme, in denen ein Zustand über einer Menge von Objekten bestimmte Constraints oder Limitierungen erfüllen muss. CSP's werden dabei als Ansammlungen von endlichen Constraints über Variablen definiert, die über bestimmte *constraint satisfaction* Methoden (Bsp. Backtracking) gelöst werden.

1.5 Arc Consistency, Kantenkonsistenz

Die Kantenkonsistenz ist eine Möglichkeit CSP's zu lösen. Dabei wird zuerst der CSP-Graph erstellt, der die verschiedenen Variablen durch binäre Constraints verbindet. Anschließend wird jede Variable überprüft, ob für jeden Wert aus der gegebenen Domäne Werte der anderen Variablen gefunden werden können, die die Constraints verbinden. Ist dies nicht der Fall, so wird der Wert aus der Domäne gelöscht. Ist schließlich kein Konflikt mehr vorhanden spricht man von *Kantenkonsistenz*.

Für Graphen in Kantenkonsistenz kann schließlich sehr einfach eine valide Variablenbelegung gefunden werden.

1.6 Handlungsplanung & STRIPS

STRIPS (Stanford Research Institute Problem Solver) stellt eine formale Möglichkeit dar Probleme, die logisch formuliert werden können, zu lösen.

STRIPS erzeugt dabei ein Vokabular $\mathcal{S}$ in Prädikatenlogik (Prädikate und Funktionssymbole) und stellt Atome $\Sigma_\mathcal{S}$ bereit. STRIPS-Zustände S stellen schließlich eine Teilmenge von $\Sigma_\mathcal{S}$ dar und beschreiben eine complete theory dar (per CWS, also eine Closed World Assumption).

Planungsaufgaben können nun also als 4-Tupel $\Pi = \langle \mathcal{S}, \mathbf{O}, \mathbf{I}, \mathbf{G} \rangle$ dargestellt werden, wobei

- **O** eine Menge an Operatoren ist
- $\mathbf{I} \subseteq \Sigma_\mathcal{S}$ den Anfangszustand beschreibt
- $\mathbf{G} \subseteq \Sigma_\mathcal{S}$ die Zielspezifikation ist.

1.6.1 Operatoren

Operatoren in STRIPS sind 3-Tupel $o = \langle \mathit{para}, \mathit{pre}, \mathit{eff} \rangle$. Dabei gilt:

- $\mathit{para} \subseteq \mathbf{V}$
- $\mathit{pre} \subseteq \Sigma_{\mathcal{S},\mathbf{V}}$
- $\mathit{eff} \subseteq \Sigma_{\mathcal{S},\mathbf{V}} \cup \neg\Sigma_{\mathcal{S},\mathbf{V}}$ (Elementweise Negierung)

- Alle Variablen in *pre* und *eff* sind in *para* aufgeführt.

Jede Operation hat also eine Vorbedingung, die erfüllt sein muss, um sie auszuführen. Der tatsächliche Effekt, der durch die Ausführung entsteht ist in *eff* definiert und wird in positive (eff^+) und negative (eff^-) Teileffekte eingeteilt.

1.7 Progression Planning: Forward Search

Bei Progression Planning wird beim Ausgangszustand begonnen. Es wird nun nicht-deterministisch eine anwendbare Aktion ausgewählt und der Folgezustand berechnet. Der Plan δ wird um die Operation erweitert und ein Test wird ausgeführt, um herauszufinden, ob ein Zielzustand erreicht ist. Ist dies nicht der Fall, wird erneut nicht-deterministisch gewählt und das Prozedere fortgeführt, bis ein Ziel erreicht ist.

Anstatt nicht-deterministisch zu wählen kann selbstverständlich auch eine Heuristik, oder allgemein eine Strategie verwendet werden.

1.7.1 Regression Planning: Backward Search

Das Regression Planning stellt die Umkehrung des Progression Plannings dar. Anstatt beim Initialzustand zu starten, wird am Ziel begonnen. Es wird dann eine Aktion gesucht, deren Effekte der Information des Zielzustandes entsprechen und der Folgezustand über die Precondition der Operation gebildet.

Hier kann anstatt von nicht-deterministischen Mitteln ebenfalls eine Strategie verwendet werden.

1.8 Relaxierung

Bei der Relaxierung werden negative Effekte gestrichen. Daraus resultiert ein deutlich einfacher zu berechnendes Planungsproblem, welches als Heuristik verwendet werden kann.

1.9 FF-System

Variante des Planungsgraphen, bei der die Zieldistanzen nach jedem Suchzustand über Relaxierungen neu berechnet werden. Es werden allerdings nur Nachfolgezustände berücksichtigt, die über die relaxierte Lösung auch generiert würden.

Die Strategie stellt dabei *Enforced Hill-Climbing* dar.

1.10 Bellman-Equation und Utility

Die Bellman-Equation

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s')U(s')$$

wird verwendet, um die Utility eines bestimmten Zustandes zu bestimmen. Sie berechnet dabei also die maximalen Ausbeuten, die erreicht werden können in Bezug auf Folgezustände und Übergangswahrscheinlichkeiten und verrechnen diese mit Discountwert γ und dem direkten Reward $R(s)$ des Zustandes selbst.

1.11 Value/Policy Iteration

Value Iteration ist ein Verfahren, um Markov-Entscheidungsprobleme zu lösen. Dabei wird für jeden Zustand der Utility-Wert errechnet. Irgendwann konvergieren die so errechneten Nutzen-Werte der Zustände gegen einen Grenzwert, der den optimalen Nutzen beschreibt.

Policy Iteration auf der anderen Seite errechnet sich anfangs über ein Gleichungssystem einen Grund-Nutzen für jeden Zustand auf einmal (Policy-Evaluation), um dann wiederholt zu überprüfen, ob die aktuelle Aktion jeden Zustandes immer noch die sinnvollste ist. Gestartet wird dabei mit einer zufällig ausgewählten Start-Policy.

2 Entscheidung

2.1 Information Gain

$$Gain(a) = I(\frac{p}{p+n}, \frac{n}{p+n}) - R(a)$$

wobei I den *Information Content* darstellt, also den Abstand zur endgültigen Aufteilung beschreibt, und $R(a)$ den *Average Information Content* nach Wahl des Attributs a repräsentiert:

$$R(a) = \sum_v \frac{p_i + n_i}{p+n} I(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i})$$

$$I(x, y) = x \ln(x^{-1}) + y \ln(y^{-1})$$

Gewählt werden sollte schließlich das Attribut, das den Gain maximiert.

2.2 Decision Trees

Entscheidungsbäume sind Möglichkeiten eine Menge von bekannten Daten so einzuteilen, das anhand der so erhaltenen Beispiele, neue Objekte in die eine oder andere Kategorie eingeteilt werden können.

Decision Trees werden dabei im besten Fall über die oben genannte Information-Gain Methode aufgebaut, also stets das Attribut zum Teilen verwendet, das den Gain maximiert. Prinzipiell ist allerdings jeder Baum, der eine Menge von Objekten klar in die Gruppen aufteilt, ein Entscheidungsbaum.

Dadurch beschreibt jeder Entscheidungsbaum also indirekt eine logische Funktion/Formel, anhand derer die Elemente klassifiziert werden.

3 Logik

3.1 Resolution & DPLL

Resolution ist eine Möglichkeit, Unerfüllbarkeit von Formeln zu beweisen. Dabei wird die Aussagenlogische Formel in CNF überführt (PL1-Formeln in Skolemform, die Quantoren ignoriert $\rightarrow$ Formel in CNF) und anschließend in Klauselform transformiert. Im nächsten Schritt können dann Resolventen gebildet werden (entweder direkt, oder durch Substitution), sodass sich am Ende die leere Menge ergibt (keine Aussage), oder eine leere Klausel übrig bleibt $\rightarrow$ Unerfüllbarkeit!

Alternativ dazu wurde die DPLL-Methode entwickelt, die die Resolutionsmethode in *Unit propagation rule*, also die Verwendung einer Klausel, die aus einem einzigen Literal besteht, oder *Splitting Rule*, also die Festsetzung einer Variable entweder auf True, oder False. Splitting Rule ist also eine Form des Backtrackings, der die Resolution leicht berechenbar macht.

3.2 PNF, SNF

Pränexe Normalenform ist eine prädikatenlogische Formel, in der sämtliche Quantoren am Anfang stehen. Vorgehensweise:

1. Doppelt vorkommende Variablen umbenennen
2. Negierungen so weit wie möglich durchreichen
3. Quantoren an den Anfang schreiben

Skolem Normalenform ist eine prädikatenlogische Formel, in der sämtliche Existenzquantoren eliminiert werden und alle Allquantoren am Anfang stehen. Vorgehensweise:

1. Formel in PNF bringen
2. Variablen, die durch Existenzquantoren gebunden sind, durch Funktionssymbole ersetzen. Parameter dieser Funktionssymbole sind die Variablen der vorhergehenden Allquantor-Variablen.

3.3 Herbrand

3.3.1 Herbrand-Universum

Das Herbrand-Universum ist die Grundmenge, die der Herbrand-Struktur zu Grunde liegt. Sie wird folgendermaßen aufgebaut:

1. Konstanten der Formel in die Menge aufnehmen
2. Iterativ alle neuen Kombinationen aus bereits gefundenen Termen der Menge und den vorhandenen Funktionssymbole in die Menge aufnehmen.

Das Herbrand-Universum ist also eine unendliche Menge von Termen.

3.3.2 Herbrand-Struktur

Herbrand-Struktur ist eine Terminterpretation, die auf Basis einer Formel definiert ist. Dabei wird das Universum gebildet und anschließend allen Variablen Terme aus der resultierenden Universum-Menge zugewiesen.

3.4 Unifizierung

Nicht übereinstimmende Terme identifizieren und versuchen eine Substitution zu finden, durch die sie äquivalent werden.

4 Stochastik

4.1 Bayes

$$P(X|Y) = \frac{P(X \wedge Y)}{P(Y)}$$

$$P(X \wedge Y) = P(Y|X)P(X)$$

$$P(X|Y) = \alpha P(X \wedge Y)$$

4.2 Normalisierung

Die Normalisierung ist eine Möglichkeit, um von relativen Wahrscheinlichkeiten auf ihre absoluten zu kommen:

$$P(X) = \alpha\langle 0.4, 0.3\rangle = \frac{1}{0.4 + 0.3}\langle 0.4, 0.3\rangle = ...$$

4.3 Bayes'sche Netze

Möglichkeit um logische Zusammenhänge in stochastischen Umgebungen effizient zu repräsentieren. Anstatt eine riesige Joint Probability-Tabelle zu verwenden, werden in den Netzen die Informationen in Ereignisse geteilt und diese über Kanten zu einem Graphen verbunden. Jeder Knoten repräsentiert dann im Prinzip eine Teiltabelle.

Inferenz kann schließlich über eine Summierung über alle versteckten Variablen und anschließende Berechnung der Gesamtwahrscheinlichkeit durchgeführt werden.

Der Vorteil gegenüber der Joint-Probability-Tabelle ist Effizienz in der Speicherung und Wartung. Nachteilig ist die Inferenzgeschwindigkeit, da Werte aus der Tabelle einfach abgelesen werden könnten.