Prüfungsprotokoll

Künstliche Intelligenz, Wissensrepräsentation, Logikprogrammierung

Datum: 16.09.2004
Zeit: ca. 35 Min
Prüfer: Prof. Ph.D. Lakemeyer
Beisitzer: Dr. Vazha Amiranashvili
Note: 1,0

Künstliche Intelligenz:

L: OK, sie haben sich ja mit Suche beschäftigt, was ist ein Zustand bei der Suche?
D: Abstrakt gesehen ist ein Zustand bei der Suche ein Knoten im Suchbaum. Dieser beschreibt dann wiederum eine Situation des Suchproblems. Neben den Knoten gibt es im Suchbaum noch die Kanten, die Aktionen repräsentieren und einen Zustand in einen anderen transformieren. Die Aufgabe der Suche ist dann das Auffinden einer Sequenz von Aktionen, die den Startzustand in einen Endzustand transformiert.
L: Und was braucht man noch bei der Suche?
D: Mhh, einen Start- und einen Zielzustand, bzw. eine Zielzustandsmenge.
L: Ja, das schon, aber noch eine andere Sache!
D: Einen Goal-Test? Also einen Test, ob ein Zustand zur Menge der Zielzustände gehört?
L: Ja, aber Sie benötigen noch Kantenkosten.
D: Natürlich!
L: Wozu wird die Suche in der KI, z.B. verwendet?
D: Zum Beispiel bei Spielen, die ein spezielles Suchproblem darstellen, oder bei der Pfadplanung.
L: Genau. Sie haben verschiedene Klassen von Suchproblemen kennen gelernt, können Sie diese nennen?
D: Single-State, Multiple-State, Contingency und Exploration Probleme genannt und erklärt. Bei Contingency-Problemen wieder auf den Zusammenhang zu Spielen hingewiesen (alle Spiele sind Contingency-Probleme, weil ...).
L: Was ist A*-Suche?
D: Erklärt wie A*-Suche funktioniert, dabei auch den Begriff der Heuristik kurz erläutert.
L: Wann ist A*-Suche optimal?
D: A* ist immer vollständig und optimal unter der Vorraussetzung, das die Heuristik nicht überschätzt, d.h. dass für alle Knoten n gilt, dass h(n)<=h*(n) ist, wobei h*(n) die tatsächlichen Kosten vom Knoten n zum Ziel dennotiert.
L: Wie sieht es mit der Komplexität von A* aus?
D: A*-Suche basiert ist ja eine Kombination von Greedy-Search mit Uniform-Cost-Search und ist daher im Worst-Case so schlecht wie Uniform-Cost-Search, also Zeit + Platzkomplexität liegen in O(b^d). Im Average Case ist diese Komplexität jedoch in der Regel deutlich besser.
L: Es gibt bezüglich der Platzkomplexität eine Variante von A*, die besser ist, kennen Sie diese?
D: Ja, IDA*! IDA* ist eine informierte Variante der Iterative-Deepening Strategie, verwendet jedoch anstatt des Tiefenlimits ein f(n)-Kostenlimit. Im Gegensatz zu A* verringert sich dabei die Platzkomplexität auf O(bd), während die Zeitkomplexität in derselben Komplexitätsklasse, also O(b^d) bleibt.
L: Kommen wir mal zur Planung, was ist beim POP-Planning den der Suchraum?
D: Der Suchraum beim POP-Planning ist die Menge der STRIPS-Operatoren!
L: Nein, nicht ganz, überlegen Sie noch mal, was Suchen Sie den beim POP-Planning?

D: Ach ja, ich suche partiell-geordnete Pläne, also ist der Suchraum der Raum aller möglichen partiell geordneten Pläne, in dem ich durch hinzufügen oder löschen eines STRIPS-Operators und der partiellen Ordnung von einem Plan in den anderen komme.
L: Genau! Wie sieht das beim Planen im Situationskalkül aus? (KR Frage!!)
D: Situationskalkül kurz erläutert und dann gesagt, dass ein Plan eine Menge von Grundaktionen ist, die die Initialsituation in eine Endsituation überführt. Da jedoch jede Folge von Aktionen eine Situation von der Initialsituation determiniert, ist der Suchraum die Menge der Situationen.
L: Ja Genau! Bevor wir jedoch zur KR kommen, gehen wir zunächst erst einmal zurück zur KI. Wir haben ja induktives Lernen betrachtet, was ist das?
D: Erklärt.
L: Wir haben auch verschiedene Verfahren zum induktiven Lernen gehabt, welche kennen Sie?
D: Neuronale Netzwerke, Decision-Trees und Decision-Lists.
L: Für welche Art von Problemen würden Sie Decision-Trees einsetzen?
D: Bei attributbasierten Problemen, wo sich die Samples gut durch die einzelnen Attribute trennen lassen. (kleiner Exkurs in die Informationstheorie)
L: Ja, was sind denn die Vor- bzw. Nachteile von Decision Lists bzw. Decision Trees?
D: Decision Trees sind besonders transparent, d.h. es wird schnell klar, wie entschieden wird. Decision Lists sind weniger transparent haben dagegen aber zwei Vorteile. Zum einen werden die Probleme kompakter dargestellt, zum anderen sind bei einer eingeschränkten Darstellung durch k-DL's weniger Samples notwendig, damit die Hypothese in einer Epsilon-Umgebung der optimalen Funktion liegt, wie man beim PAC-Learning sieht.
L: Aber PAC ist doch kein Kriterium für Decision Lists, sondern ein allgemeines Kriterium für das Lernen. Was hat das mit DL's zu tun?
D: PAC Learning ist ein Kriterium dafür wie viele Samples mindestens benötigt werden, damit eine Hypothese mit W'keit (1-d) in einer Epsilon Umgebung der optimalen Funktion liegt. Dabei ist dieses Verfahren neben d und Epsilon von der Größe des Hypothesenraums abhängig. Und dieser ist bei einer eingeschränkten Darstellung durch k-DL's eben kleiner als bei unbeschränkten DT's.
L: Wie groß ist der Hypothesenraum den bei k-DL's?
D: Der liegt in O(n^k * Log2(n^k)).
L: Nein, das stimmt nicht, der Hypothesenraum ist immer irgendetwas mit 2^..
D: Ach ja, die Formel lautet ja ln(|H|), daher ist der Hypothesenraum bei k-DL's natürlich: 2^(O(n^k ....).
L: Genau, was ist denn der BIAS beim DT-Learning?
D: Die Einschränkung des Suchraums durch die DT-Learning Strategie. (DT Learning ganz kurz erklärt).
L: Mhh, ja, aber welche Samples werden denn dann betrachtet?
D: ???
L: Sie haben es ja schon quasi gesagt!
D: (Wann habe ich irgendwas gesagt??)
L: Die kürzesten!
D: Ach ja! (Geschickt so getan, als hätte mir das die ganze Zeit auf der Zunge gelegen).
L: Was ist das Bayes Update und wo wird es in der KI verwendet?
D: Das Bayes-Update ist ein Verfahren welches auf einer Unabhängigkeitsannahme und der Bayes Regel beruht. Es wird verwendet um zu einer bestehenden Wahrscheinlichkeit Evidences hinzuzufügen. In der Praxis benutzt man das Bayes Update zum Beispiel in der Lokalisierung. Hier wird zunächst eine W'keit P(L | s1) berechnet, also die Wahrscheinlichkeit dafür dass man sich an Position L befindet unter der W'keit, dass man Sensorreading s1 hat. Nun haben wir ein anderes Sesorreading s2 zur Verfügung und möchten die W'keit für P(L | s1 und s2) berechnen. Dafür verwendet man dann das Bayes Update.

L: Und was muss dafür gelten?
D: Die Evidences müssen stochastisch unabhängig sein.
L: Im eben genannten Beispiel bedeutet das was genau?
D: Dass P(s1 l L und s2) = P(s1 l L) ist. Diese Vereinfachung ist notwendig, da sonst die Berechnung von P(s1 l L und s2) sehr schwierig ist.
L: Wie ist das in der Praxis mit der stochastischen Unabhängigkeit?
D: In der Praxis werden Sensorreadings als stochastisch unabhängig angenommen, die an derselben Position gemacht werden. Das stellt natürlich eine Vereinfachung dar, funktioniert jedoch trotzdem sehr gut.
L: (Irgendetwas dazu erläutert, kann mich aber nicht mehr erinnern). Sie kennen ja auch die Markov Annahme, erklären Sie diese!
D: Die Markov Annahme ist eine Annahme darüber, das die Welt deterministisch ist, d.h., dass jede Veränderung der Welt Resultat von Aktionen ist.
L: Und was macht die Markov-Annahme noch?
D: Mhh, sie nimmt an, dass es keine Störfaktoren, d.h. keine dynamischen Elemente in der Welt gibt???
L: Sie betrachtet nur die letzte Situation!
D: Ach ja! (Wieder mal so getan, als wäre mir das nur entfallen).
L: Und wie ist das in der Praxis?
D: Bei leichten Abweichungen funktioniert die Markov Annahme noch so, bei großen Abweichungen müssen die Fehler herausgerechnet werden.

Wissensrepräsentation:

L: Sie haben Beschreibungslogiken kennengelernt. Warum verwendet man diese?
D: Bei der FO-Logik wird Information sequentiell repräsentiert, z.B. also Stundet(xyz), Martikelnummer(xyz, 222222),... Dabei ist es aber recht schwierig alle relevanten Informationen über ein Objekt, also eine Person zu sammeln. Dafür verwendet man bei den Beschreibungslogiken komplexe Prädikate, sogenannte Descriptions, die alle relevanten Information über ein Objekt vereinen. Ein weiterer Grund für die Verwendung von Beschreibungslogiken ist die effiziente Berechnung von Inferenzen, also zum Beispiel Subsumierung (umgangssprachlich beschriben) und ob ein Individuum zu einem Konzept gehört, also ob gilt i Element C.
L: Das gilt natürlich nur für eingeschränkte Description Logics!
D: Ja natürlich, für FL-, also FL ohne den RESTR-Operator.
L: Was können Sie mir zur Interpretation von Description-Logics erzählen?
D: Da die Semantik der Description-Logics wie die der FO-Logik aus der Logik herausführt benötigt man eine Struktur, bzw. eine Interpretation in der man die Logik Interpretiert. Diese besteht aus einem Tupel, (D,Phi), ...
L: Wie ist der AT-LEAST Operator formal definiert?
D: Phi(AT-LEAST n r)= ...
L: Was ist eine Konzept-Hierarchie?
D: Eine Konzept-Hierarchie ist eine Hierarchie auf Konzepten. Sie wird verwendet, damit Anfragen, wie z.B. welche Konzepte von einem Konzept N subsumiert werden keine Suche über die gesamte KB benötigt, sondern nur über die relevanten Objekte in der Hierarchie.
L: Wie erstellt man eine solche Hierarchie?
D: Gehen wir von einer bestehenden Hierarchie T aus und fügen ein neues Konzept N hinzu. Dann berechnen wir zunächst alle Konzepte die N subsumieren, ...#
L: Wir hatten ja auch die nicht-monotonen Logiken, was ist das?

D: Kurz die Idee des Non-Monotonic Reasoning erklärt und dabei ein bisschen die Closed Worl Assumption angeschnitten.
L: Sie haben ja auch inheritance Networks kennengelernt. (Hat eines aufgemalt). Können Sie dieses in autoepisdemic Logic übersetzen?
D: Gemacht.
L: Wie viele stabile Expansionen hat die KB?
D: Zwei!
L: Wie berechnet man denn die Erweiterungen?
D: Algorithmus erklärt, also 2^n Belegungen durchtesten und...
L: Ist dann der Plaptyrus ein Eierleger, oder nicht? (Irgendwie so in die richtige Richtung gelenkt)
D: Kurz noch mal die Definition der Expansion wiederholt und mich dann auf nur eine Expansion festgelegt. Erklärt, warum es nur eine gibt und das auch noch mal mit der preffered Extension des Inheritance Networks gefestigt.
L: Genau! Wenn es nämlich nur eine preffered extension gibt, dann gibt es auch nur eine stabile Expansion (oder so).
L: Sie haben auch abductive Reasoning gehabt, was ist das?
D: Erklärt!
L: Können Sie ein Beispiel nennen, wo abductive Reasoning gebraucht wird.
D: In einer vereinfachten Form der Diagnose...
L: Und woraus besteht da die KB?
D: Aus Implikationen der folgenden Form: diseases und conditions => symptoms
L: Ja, das stimmt. Wir haben auch ein Beispiel kennengelernt, wo es um die Diagnose von Schaltkreisen ging. Können Sie das Erklären?
D: Schaltkreisbeschreibung und Fehlermodell als Logische Formeln in der KB, Berechnung von Primimplikanten, ... (alles erklärt)
L: Was ist ein Primimplikant?
D: Eine Klausel c ist Primimplikant, wenn gilt, ...
L: Und warum benötigt man die Primimplikanten um das Fehlermodell zu berrechnen?
D: Weil man Erklärungen für das Verhalten des Schaltkreises sucht. Also sucht man z.B. eine Erklärung für ein Literal l, such man einfach nach der Menge der (P-l) so dass P Primimplikant, ...
L: Und sind die Primimplikanten leicht zu bestimmen?
D: Nein, das leider nicht, weil Primimplikanten durch Resolution berechnet werden (kurz erklärt wie)

Logikprogrammierung:

L: Sie haben ja schon einige Male über Resolution geredet, können Sie mir kurz erklären, was Resolution ist?
D: Resolution ist ein Verfahren um die Unerfüllbarkeit einer Formelmenge zu bestimmen. Dabei gilt die Äquivalenz: A ist unerfüllbar gdw. Man kann aus A die leere Klausel resolvieren.
L: Wie funktioniert denn ein Resolutionsschritt?
D: Im AL-Fall wie folgt: Nehmen wir an, wir haben zwei Klauseln {L U C1} und {L1 U C2} mit L = nicht L1 ....
L: Genau, aber warum können Sie eine Klausel so einfach „wegresolvieren“?
D: Das geht, weil Resolventen Folgerungen der Eingabeklauseln sind und...
L: Ja. Wie komplex ist Resolution?
D: Im eben vorgestellten AL-Fall exponentiell entscheidbar. Im FO-Fall semi-Entscheidbar, das heißt ...
L: Und wie geht ein Resolutionsschritt im FO-Fall?

D: Erklärt und dabei zunächst Skolemisierung, und dann den Unifikationsalgorithmus umrissen.
L: Wie komplex ist Unifikation?
D: Erklärt, das es lineare Algorithmen für Unifikation gibt, der aus der Vorlesung jedoch exponentiell war.
L: Warum war der aus der Vorlesung exponentiell?
D: Eine Instanz skizziert, in denen der der Ausdruck exponentiell in der Anzahl der Variablen wurde und auch erklärt, dass man diesen Fall mit einem Algorithmus mit einer Graphenähnlichen Datenstruktur linear verarbeiten kann.
L: Und warum verzichtet Prolog dann auf den Occur-Check? (Ach ja, ich habe ihm irgendwann vorher erzählt, dass das occur-check heißt).
D: Laut Herrn Ivan aus optimierungsgründen...
L: Mhh, der Algorithmus, der in der Vorlesung dran kam ist auch der, den Prolog verwendet und daher wird auf diesen occur check meist verzichtet. (War hier eher eine Diskussion als eine Prüfungsfrage)
L: Wir haben ja gesehen, dass Resolution im Allgemeinen schwer ist, was kann man tun, damit Resolution besser wird?
D: Zum einen kann man eingeschränkte Logiken, wie z.B. Horn Logik und SLD Resolution verwenden (SLD-Resolution kurz erklärt).
L: Und wie komplex ist SLD Resolution?
D: Im FO-Fall immer noch exponentiell, aber dafür viel ziel gerichteter als normale Resolution, also im average Case deutlich besser als normale Resolution, ...
L: Welche Dinge können Sie mit Horn-Logik darstellen?
D: Horn-Logik bildet eine echte Einschränkung, z.B. kann man unsicheres Wissen, d.h. alpha oder beta nicht darstellen.
L: Aber was können Sie denn genau darstellen? (Hat mich dann auch ganz direkt auf müh-rekursiv Aufzählbare Funktionen angesprochen)!
D: Habe Herrn Lakemeyer darauf hingewiesen, dass ich zwar in etwa weiß worauf er hinaus möchte, aber der Bereich der müh-rekusiven... nicht im letzten Skript vorkam. (Da sie ja wirklich nicht im letzten Skript stehen, hat Herr Lakemeyer mir das durchgehen lassen, aber ich würde jedem Empfehlen, sich diesen Bereich im vorletzten Skript anzugucken).
L: Und was gibt es noch für Möglichkeiten Resolution besser zu machen?
D: Ordering Goals, Rule Formulation, Controlling Backtracking durch CUT,... erklärt
L: Erklären Sie mal den CUT-Operator.
D: Syntax und Semantik des CUT Operators, sowie ein Beispiel erklärt.
L: Tja, da ist die Zeit wohl vorbei. Würden Sie uns dann bitte zur Beratung alleine lassen?

Anmerkungen: Zunächst ist dieses Prüfungsprotokoll ein reines Gedächtnisprotokoll und wurde 1,5 Wochen nach meiner Prüfung, im Urlaub, erstellt. Sicher habe ich also einige Fragen vergessen und einige hinzugefügt - Im Großen und Ganzen müsste dieses Protokoll aber in etwa der richtigen Prüfung entsprechen.

Fazit: Herr Lakemeyer und auch Dr. Vazha Amiranashvili waren sehr angenehm als Prüfer. Wichtig ist wohl, dass Herr Lakemeyer weniger auf Formalia (d.h. auf exakte Definitionen) achtet als auf wirkliches Verständnis. So schien es mir bei manchen Fragen, als ob er erst dann zufrieden war, wenn man das Erzählte an einem kurzen Beispiel erläuterte. Meine Strategie, alles assoziierte Wissen zu einem Thema im Rahmen einer Frage zu erzählen hat sehr gut funktioniert. So habe ich beispielsweise sofort die SLD-Resolution erklärt, wenn es um eine Frage mit Horn-Formeln ging, oder auch immer praktische Anwendungen der Dinge mit erläutert (wo ich konnte). Ich stand wohl genau zwischen 1,3 und 1,0 (besonders wegen der müh-rekursiv...), und schätze, dass ich die 1,0 nur bekommen habe, weil ich immer viel mehr gesagt habe, als ich musste.