>>> Gedächtnisprotokoll <<<

Prüfer: Lakemeyer
Beisitzer: Gero Iwan
Fächer: Betriebssysteme, KI, KR
Datum: 14.04.2005
Dauer: ca. 45min
Note: 1,3

Reihenfolge wurde von Lakemeyer vorgeschlagen, war aber auch meine Wunschreihenfolge. Ich habe versucht möglichst genau aufzuschreiben wie die Prüfung gelaufen ist und auch den Stil von Lakemeyer und meinen Antworten einzufangen. Es kann sein, dass ich etwas vergessen habe oder die Reihenfolge nicht ganz stimmt. Ausserdem glaube ich, dass ich die ein oder andere Frage (F) ein wenig so umformuliert habe, wie ich sie aufgefasst habe und daher meine Antwort (A) ausgefallen ist. Wie in anderen Protokollen schon zu lesen ist: Lakemeyer versucht die Prüfung so zu gestalten, dass man selber auf alles kommt, aber lässt einen auch nicht versauern. Ich hatte auch bei den Fragen, die mir nicht so gut lagen, nicht das Gefühl mir mit der Antwort die ganze Prüfung zu versauen, da er entweder meine Punkte aufgegriffen hat oder einfach die Frage nochmal neu formuliert hat. Insgesamt also eine sehr angenehme Prüfung.
Zur Vorbereitung kann ich nur sagen, dass die etwa 8 Wochen (davon ab der zweite Hälfte mit dem ständigen Gefühl, dass die Zeit nicht reicht; eigentlich eher 9 Wochen, aber zwischendurch haben mein Lernpartner und ich auch mal einen Tag wegen zu intensiven Feierns freigemacht) für mich als Minimum anzusehen sind. Das Betriebssystemebuch ist eigentlich ganz interessant geschrieben... wenn man nicht unter Zeitdruck steht... ich hätte eine kompaktere schreibweise bevorzugt, ausserdem sollte man die Detailprobleme die erwähnt werden nicht überschätzen. Das zusammenschreiben der wichtigen Punkte und vor allem die nochmalige Zusammenfassung des zusammengeschriebenen haben mir sehr dabei geholfen den Riesenberg von Informationen in mein Hirn zu bekommen. Die KR-Vorlesung hatte ich übrigens nicht gehört und war nur mässig oft in der Übung... wer die Vorlesung gehört hat könnte da also vielleicht eine Woche vorbereitung sparen... naja, wer sich wirklich sehr gut mit der Vorlesung beschäftig hat bevor er ans lernen geht, schafft es vielleicht ja auch deutlich schneller als in besagten 8 Wochen... ;)
Jetzt aber zu den Fragen:

>>> Betriebssysteme <<<

F: Wie unterscheiden sich UNIX und Win2000 grundsätzlich?
A: In Win2000 ist die GUI ins BS integriert, UNIX ist insgesamt deutlich schlanker. Ansonsten gibts in ziemlich allen Bereichen Unterschiede (Speichermanagement, Dateisystem, ...aber nicht direkt auf Unterschiede eingegangen).

F: Schlanker würde ich als Vorteil für UNIX gelten lassen, kann man denn über Win2000 auch positives berichten?
A: Zum Beispiel im Dateisystem: Zugriffsschutz kann detaillierter eingestellt werden als in UNIX (UID, GID), Verschlüsselung unterstützt, Komprimierung.

F: Bleiben wir beim Dateisystem. Was gibt es denn da noch, was es bei UNIX nicht gibt?
A: Streams. Dateien in mehrere Teile (die Streams) aufteilen und Zugriff auf unterschiedliche Streams durch verschiedene Prozesse erlauben. (Hier bin ich nicht sicher ob ich nicht auch was über die MFT von NTFS gesagt habe.)

F: Und wofür kann man Streams verwenden?
A: Bei grossen Datenbanken können so Prozesse auf verschiedene

Tabellen zugreifen.

F: Ja, das ist ja das klassische Beispiel dafür, aber denken sie mal mehr in Richtung Multimedia-Betreibssysteme.
A: (weiss nicht mehr ganz genau was ich darauf geantwortet habe)

F: Also in einer Multimedia-Datei. Was könnten denn da die Streams sein?
A: Die einzelnen Medienspuren (Video, Audio).

F: Genau, und Untertitel und was es da sonst noch alles gibt. Kommen wir zu Video-on-Demand. Was ist denn da für den Benutzer interessant?
A: (hatte mit near-Video-on-Demand begonnen aber schnell gemerkt dass ich das Falsche erkläre) ...der Benutzer will natürlich dass der Film nicht ruckelt.

F: Also was heisst das jetzt? Wann ruckelt es nicht?
A: Der Server muss die Daten schnell genug schicken können, damit die Wiedergabe beim Benutzer nicht ins stocken gerät.

F: Wenn man also einen Film sieht, wie schnell muss der Server dann die Daten senden?
A: Wenn man ein Datenpaket pro Frame verschickt dann also etwa 25 Pakete pro Sekunde.

F: Ja, oder eher so 30, aber okay. Was kann man denn machen, wenn man weiss, dass ein Film populärer ist als andere?
A: Wenn man mehrere Filme auf einer Platte hat, dann sollte man die häufig geguckten Filme möglichst nicht an die Ränder legen um den Plattenkopf nicht unnötig weite Wege zurücklegen lassen zu müssen. Also am besten legt man die populärsten Filme in die Mitte.

F: Ja, nach dem Orgelpfeifenalgorithmus. Kommen wir zu Eingabegeräten. Nehmen wir beispielsweise mal eine Tastatur. Wie wird denn da eine Eingabe verarbeitet?
A: Also Tastaturen sind erstmal klassische zeichenorientierte Eingabegeräte im Unterschied zu Festplatten. Dann gibt es da noch den Unterschied ob die Eingabe direkt von der CPU verarbeitet wird, oder ob Escape-Sequenzen noch vorher umgesetzt werden (z.B. Backspace), aber in der Regel wird wohl letzteres gemacht (?!?).

F: Okay, also wie geht der Prozessor dann mit einer Eingabe um? Wieviele Zeichen liest man den ein wenn ich ein "a" eintippe?
A: Man liest einen Stream (den Begriff habe ich irgendwie aus Java, weiss nicht mehr genau ob es dafür nicht ein anderes Wort gab, aber scheint ihn auch nicht gestört zu haben) und wenn bei der Tastatur eine Eingabe anliegt dann erhält das Betriebssystem einen Interrupt und der Prozessor kann dann von der Tastatur die Eingabe abfragen oder der DMA-Chip übernimmt das und legt die Daten in den Hauptspeicher.

F: Und wie kommt der Buchstabe auf den Bildschirm?
A: Hmm, da gab es so eine Funktion unter Win2000 mit der man den Buchstaben, also die Bitmap-Repräsentation davon, auf den Bildschirm kopieren kann, aber keine Ahnung mehr wie die heisst. (auf weiteres Nachfragen) Der Prozesser kopiert das "a" in den Videospeicher der Grafikkarte der das ganze dann in ein analoges Signal umwandelt und an den Monitor schickt.

F: Ja, es gibt inzwischen auch Monitore die digital arbeiten, dann ist das Umwandeln nicht mehr nötig. (Irgendwie wollte er auch noch darauf hinaus, dass beim Drücken eines "a" einmal "a gedrückt" und nochmal "a losgelassen" verschickt wird.)

>>> KI <<<

F: Fangen wir mit der Suche an. Sie haben in der Vorlesung Beispiele kennengelernt bei denen die Wege zum Ziel wichtig waren und wo dem nicht der Fall war. Wo interessiert man sich denn nicht für den Weg?
A: Hmm, also Beispiele... (habs mal mit dem Strassen-A*-Beispiel versucht, aber das war nix)

F: Naja, das ist jetzt ein Beispiel wo sie sich für den Weg interessieren...
A: Ja, stimmt. Also wenn man sich nicht für den Weg interessiert, dann muss also das was man sucht auch schon im Zielknoten drin stecken. Wir hatten ja beim Planning als Suchraum die Menge aller möglichen Pläne, wobei als Aktionen das Hinzufügen von Knoten zum Plan zu sehen ist. Da steckt dann also ein kompletter Plan im Zielknoten.

F: Ja, zum Beispiel. Beim 8-Queens-Problem etwa hat man das auch. (Die anschliessende Frage weiss ich nicht mehr genau, ich habe einfach mal mit Breiten-/Tiefensuche und so weiter angefangen weil ich nicht so recht wusste wohin die Frage führen sollte... er hat mich dann auch unterbrochen und gesagt er wolle auf die "gradient descent"-Suche hinaus.)
A: Wenn man da etwa eine konvexe Fehleroberfläche hat dann konvergiert die Suche, aber wenn das nicht der Fall ist wie zum Beispiel bei Neuralen Netzen mit mehreren hidden Layern dann weiss man nicht ob man in einem lokalen oder globalen Minimum gelandet ist.

F: Und was kann man in so einem Fall machen?
A: Wie heisst es nochmal... "simulated annealing". Das funktioniert so, dass man nicht immer in die Richtung des steilsten Anstiegs geht, sondern einfach auch mal in eine zufällige Richtung. (Bin dann etwas unglücklich zum Verfahren mit den zufälligen Startpunkten gewechselt und habe die zufälligen Startpunkte wieder mit zurück ins "simulated annealing" genommen.)

F: Also das mit den zufälligen Startpunkten stimmt nicht so ganz, da gab es aber noch die Temeratur. Was fällt ihnen denn dazu noch so ein?
A: Ja, also es läuft so, dass man zu beginn viele zufällige Richtungen einschlägt und dann langsam immer öfter in die Richtung des steilsten Anstiegs geht. Man hofft dann dadurch in einem möglichst tiefen Tal zu landen.

F: Okay, kommen wir zu einem anderen Thema. Warum verwendet man denn Bayes Netze?
A: Also Bayes Netze sind eine kompakte Darstellung von Wahrscheinlichkeitsverteilungen. (Habe dann erklärt, wie man ein Neurales Netz mit unabhängigen Wahrscheinlichkeiten baut... in der Zeit hat er folgendes Netz aufgemalt: ...hoffe man kanns erkennen)

```
B<---A
|   /|
|  / |
| /  |
vv   v
C    D
```

F: Wie berechnet man hier P(A, negB, negD)?
A: (Hier bin ich etwas ins straucheln geraten... man hätte sicher nur Bayes Update anwenden müssen oder so (habs nicht mehr nachgeschaut), aber ich habe versucht über die C-Fälle zu summieren und dadurch hätte das C dann rausfallen sollen, aber ich war wohl etwas unsicher dabei, sodass er meinte ich wäre wohl

auf dem Holzweg und sollte doch nochmal versuchen neu anzufangen. Ich habe dann meinen Verdacht geäussert, dass man hier das C aus dem Netz rausschmeissen könnte wegen der Verbesserung mit der "variable elimiation" aus dem Buch, aber dass ich mich da auch irren könnten. Gero zwinkerte mir zu und flüsterte was von wegen "stimmt".)

F: Dann schreiben sie mir doch mal einfach das Ergebnis hin.
A: P(A) * P(negB|A) * P(negD|A) (...glaube ich doch.)

F: Was versteht man denn unter PAC-Learning?
A: Man möchte sicherstellen, dass in z.B. unter 5% der Fälle das Ergebnis um z.B. mehr als 5% vom richtigen Ergebnis abweicht.

F: Wieviele Beispiele muss man denn dafür trainieren?
A: Da gabs diese Formel in der die Hypothesenanzahl logarithmisch eingegangen ist.

F: Und was heisst das jetzt für boolsche Funktionen?
A: Man hat 2^{2^n} Hypothesen bei n Attributen, also benötigt man exponentiell viele Beispiele.

F: Genau, man muss also fast alle lernen.
(Ich glaube das wars zu KI.)

>>> KR <<<

F: Bei Knowledge Representation kann man Wissen in FOL notieren und dann Schlussfolgerungen daraus gewinnen. Was ist denn das Problem bei diesem Ansatz?
A: Bei FOL verwendet man im allgemeinen ja Resolution und das ist zu schwierig für KR.

F: Wie schwierig ist das denn genau?
A: Semi-entscheidbar, aber das ist bei KR ja nicht wünschenswert.
(Ein wenig rumgedruckst.)

F: Wenn ein Roboter an einer Ampel steht soll er ja schliesslich nicht ewig da stehen bleiben. Was kann man denn dagegen unternehmen?
A: Man kann die Sprache einschränken. Zum Beispiel kann man sich auf Horn-Formeln beschränken was ja Prolog so macht. Die Resolution bleibt zwar semi-entscheidbar, aber man muss nur mit den Anfangsklauseln resolvieren.

F: Und wann wird die Resolution entscheidbar?
A: Bei Prädikatenlogik. (Habe dann was von wegen Herbrand-Theorem erzählt, aber irgendwie wars das nicht so recht, aber hier kann ich mich auch irren.)

F: Wie sieht es denn da mit der Komplexität aus?
A: Exponentiell?!?

F: Nein, das geht linear.
A: Echt? Hmmm...

F: Ist schon gut, das gebe ich ihnen mal als Hausaufgabe mit. Schauen sie sich das nochmal an. Was kennen sie denn noch für Möglichkeiten die Sprache einzuschränken?
A: Wir hatten da noch Description Logics. Der Vorteil dabei ist, dass man Schlussfolgerungen über einen Matching-Algorithmus berechnen kann.

F: Wie kann man denn (ALL r C) nach FOL übersetzten?
A: Also semantisch bedeutet das "alle x, die aussschliesslich mit

C-Elementen in r-Relation stehen" (Habe auch nochmal die Interpretation von ALL aufgeschrieben). Also muss da sowas wie P(y) für das C stehen. Und habe dann gedacht man könne für das r sowas wie y = f(x) schreiben, aber Lakemeyer meinte es wäre geschickter Q(x, y) zu nehmen... stimmt! Das passt besser. Dann muss da noch "and irgendwas mit x" stehen... ach nein, Implikation wie man in der Interpretationsformel sieht. Dann noch allquantifizieren, aber zum Schluss hat er mir dann nochmal ganz kurz erklärt wie das denn jetzt genau zusammenhängt...

F: Dann jetzt mal zu Default Logics. Was ist denn das?
A: Man hat jetzt zusätzlich noch Default-Regeln. (Versucht es mit Worten zu sagen aber dann doch schnell das Tripel aufgeschrieben und es daran erklärt) ...das kann man dann als FOL-Formel schreiben: Alpha and negB... ach Mist, das ist natürlich nicht FOL sondern Autoepistemic Logic. Bei Default Logic bekommt man über die rekursive Formel mit dem Delta die Extensions.

F: (hier weiss ich nicht mehr ganz genau was er dazu gesagt hat, aber es kam dann nochmal die Autoepistemic Logic) Sie hatten ja schon fast die Übersetzung zur Autoepistemic aufgeschrieben, wie geht das denn jetzt genau?
A: Alpha and negBnegBeta implies Gamma

F: Und glauben sie jetzt, dass Default Logics und Autoepistemic Logics sprachäquivalent sind?
A: Also von Default Logics nach Autoepistemic sollte auf jeden Fall gehen, aber an den Umkehrschluss glaube ich nicht (konnte das aber nicht recht begründen).

F: (ohne lange zu warten hat er gesagt warum der Umkehrschluss tatsächlich nicht geht, aber das weiss ich ehrlich gesagt nicht mehr genau... dann hat er mich gebeten kurz vor die Tür zu gehen.)

Tja, das wars... na dann mal viel Erfolg!