Universität Dortmund
Katharina Morik
Paul Fischer, Ralf Lübeck, Arnulf Mester

Dienstag, 25.4.95
Abgabe bis Dienstag, 2.5.93 um 12 Uhr
Briefkästen 17 und 18 in Pavillon 6

Übungen zur Stammvorlesung
Künstliche Intelligenz
SoSe 1995
Blatt 0

AUFGABE 0.1 (Kernaufgabe) :
Im Verzeichnis `ki000/U0` liegt eine kleine Datenbank `stamm.pl`, in der Verwandtschaftsverhältnisse beschrieben sind. Die angegebenen Fakten sind von der Form

`istKindVon(maria,otto)` bedeutet, daß Maria ein Kind von Otto ist.

Außerdem sind die Geschlechter der Personen angegeben

`mannl(otto)` bedeutet Otto ist ein Mann, `weibl(kim)` bedeutet Kim ist eine Frau,

Benutzen Sie bei den folgenden Aufgaben auch den Debugger, speziell dessen Trace-Funktion, um zu sehen, wie Prolog die Datenbank durchsucht. Falls Sie die Prädikate `not` bzw. `member` benötigen, müssen Sie sie mit `?- [library(not)].` bzw. `?- [library(basics)].` bereitstellen.
a) Definieren Sie weitere Familienverhältnisse wie `istBruderVon`, `sindGeschwister`, `istVaterVon`, `sindElternVon` usw. Planen Sie vorher, in welcher Reihenfolge dies geschieht, um einfachere Definitionen für kompliziertere benutzen zu können. Testen Sie `sindElternVon(monika,otto,maria)` (Monika und Otto sind die Eltern von Maria) und `sindElternVon(kurt,otto,sabine)`. Es sollte einmal YES und einmal NO herauskommen.
b) Definieren Sie `istNachfahreVon` und testen Sie `istNachfahreVon(tim,otto)` und `istNachfahreVon(friederich,tim)`. Finden Sie alle Nachfahren von `otto`.
c) Überlegen Sie sich eine Plausibilitätsprüfung für die Datenbank. D.h. einen Test, ob die Datenbank unmögliche Familienverhältnisse widerspiegelt.
d) Definieren Sie `istOpaVon` und überlegen Sie sich, wo dort ein CUT eingeführt werden könnte, um unnötiges Suchen zu vermeiden.
AUFGABE 0.2:
Im Verzeichnis `ki000/U0` liegt eine kleine Datenbank `graph.pl`, in der die Repräsentation eines (zusammenhängenden) Graphen abgespeichert ist. Er hat die Knoten `a,b,c,...,z`.

`kante(a,b,10)` bedeutet, daß eine Kante mit Gewicht 10 zwischen `a` und `b` existiert.

Wir gehen hier von einem ungerichteten Graphen aus, auch wenn wir nur `kante(a,b,10)` angeben und nicht auch noch `kante(b,a,10)`. Weiterhin gibt es ein Programm `durchlauf`, das den Graphen durchsuchen soll, d.h. alle Kanten abläuft und alle Knoten besucht. Für diese Aufgabe können die Gewichte ignoriert werden.
a) Überlegen Sie sich, was das Programm macht und führen Sie einen Testlauf mit `durchlauf(a,X)` durch. Analysieren Sie, warum nicht alle Knoten erreicht werden.
b) Verbessern Sie das Programm so, daß es alle Knoten besucht und terminiert.
HINWEISE: Benutzen Sie Listen. Beim Durchlauf kann ein Knoten durchaus mehrfach besucht werden.

Abgabe von Programmen: bitte **dokumentiertes ASCII-File** per UPS (Dateiname = Nachname.Gruppennummer) an den jeweiligen Übungsleiter, siehe unten. Es muß ein Beispiel für den Testaufruf enthalten. Die Datei `graph.pl` enthält ein Beispiel. Andere Abgaben bitte in die Briefkästen mit **Gruppennummer** oder **Terminangabe**.

Name	Raum	Sprechzeiten	Telefon	Account
Paul Fischer	IV-332	wenn anwesend	4684	ki001
Arnulf Mester	V-509	Mo 13:30 und n.V.	4662	ki002
Ralf Lübeck	V-431	wenn anwesend	4746	ki003