

## **Vorschläge für Projektaufgaben**

### **1. Studienplanung**

Entwerfen und Implementieren Sie ein Programm zur individuellen Studienplanung unter Berücksichtigung der folgenden Anforderungen:

- Das Programm soll auf der Grundlage der Angaben zu den
  - Grundlagenveranstaltungen,
  - Projekten,
  - Vertiefungsvorlesungen,
  - und Seminaren

des Hauptstudiums Informatik aus dem Lehrtableau für das laufende Semester einen individuellen Wochenplan zusammenstellen.

- Bei der Planung soll das Programm zumindest die folgenden Vorgaben berücksichtigen:
  - die maximale Präsenzzeit pro Woche,
  - das gewählte Vertiefungsgebiet,
  - die minimale bzw. maximale Anzahl an Grundlagenveranstaltungen, sowie
  - die minimale bzw. maximale Anzahl von Vertiefungsveranstaltungen.

Diese Angaben können ggf. auch interaktiv erfragt werden.

- Das Planung sollte durch eine Reihe von Präferenzen steuerbar sein, z.B.
  - die Notwendigkeit zur Fortsetzung eines im vorangegangenen Semesters begonnenen Projekts ,
  - die Wahl eines ein- bzw. zweisemestrigen Projekts,
  - die Angabe von Zeitslots, die möglichst nicht verplant werden sollen usw.

Diskutieren Sie Lösungsalternativen und die Vorteile der von Ihnen gewählten Repräsentationsmittel und Problemlösungstechniken.

Erarbeiten Sie Vorschläge, wie der Leistungsumfang Ihres Programms durch zusätzliche Funktionalität erweitert werden kann.

## 2. Dialogsysteme

Implementieren Sie ein System zu natürlichsprachlichen Kommunikation mit einer deduktiven Datenbank. Ihr System soll natürlichsprachliche Aussagen in Datenbankeintragen umwandeln und dabei die Datenbank auf Konsistenz überprüfen. Natürlichsprachliche Anfragen sollen ggf. unter Zuhilfenahme von Schlußfolgerungen beantwortet werden. Beschränken Sie dabei die zulässigen Äußerungstypen auf wenige Konstruktionen, z.B.

**N:** Hans ist ein Student.

**S:** Habe verstanden: `instance_of(hans,student), prop(hans,maennlich)`

**N:** Susi ist eine Studentin.

**S:** Habe verstanden: `instance_of(susi,student), prop(susi,weiblich)`

**N:** Wer ist ein Student?

**S:** Hans ist ein Student.

**N:** Was ist Hans?

**S:** Hans ist Student und Hans ist männlich.

**N:** Susi ist ein Mann.

**S:** Vorhandene Information: *Susi ist weiblich*. Neue Information ignoriert.

**N:** Susi ist wirklich ein Mann.

**S:** Habe verstanden: `prop(susi,maennlich)`. Alte Information gelöscht.

**N:** Studenten sind normalerweise faul.

**S:** Habe verstanden: `instance_of(X,student), not(abnormal(X)) -> prop(X,faul)`

**N:** Wie sind Studenten?

**S:** Studenten sind normalerweise faul.

**N:** Wie ist Hans?

**S:** Hans ist faul.

**N:** Susi ist fleißig.

**S:** Habe verstanden: `prop(susi,fleissig)`

**N:** Wie ist Susi?

**S:** Susi ist fleißig.

**N:** ...

Mögliche Erweiterung: Generieren Sie auf Anfrage auch natürlichsprachliche Begründungen.

### 3. Handlungsplanung

Implementieren Sie auf der Grundlage der auf der Webseite für das Lehrbuch angegebenen Programmfragmente drei verschiedene Planungsalgorithmen. Testen und vergleichen Sie Ihre Implementierungen mit dem Vorwärtsplaner aus der Übung hinsichtlich Vollständigkeit, Korrektheit und Effizienz. Axiomatisieren Sie dazu die Klötzchenwelt und konstruieren Sie für Ihre Experimente unterschiedlich komplexe Planungsprobleme. Klassifizieren Sie Ihre Probleme nach dem Aufwand für das Auffinden einer Lösung. Was charakterisiert schwierige Planungsprobleme?

Modellieren Sie eine nichtriviale Planungsaufgabe aus Ihrem alltäglichen Umfeld und testen Sie sie mit Ihren Planungsprogrammen. Ein Beispiel für eine Planungsaufgabe könnte etwa das Vorbereiten des Frühstücks sein, einschließlich Tisch decken, Kaffee aufbrühen und eingießen, sowie Vorbereiten und Belegen des Toastbrotens.

Hinweis: Für die Lösung dieser Aufgabe benötigen Sie Sictus-Prolog, da SWI-Prolog nicht die für das zielgesteuerte Planen erforderlichen Coroutinen anbietet. 30-Tage Evaluationslizenzen sind unter <http://www.sics.se/isl/sicstuswww/site/eval.html> erhältlich.

#### Hinweise zur Bearbeitung:

1. Stellen Sie die Konzeption Ihrer Lösung in einem Kurzvortrag in der 12. Semesterwoche am 17. oder 20.1.2005 vor.
2. Beschreiben Sie die Grundlagen Ihrer Implementierung in einem max. 15-seitigen Bericht, der bis zum 28.1.2005 (als .pdf-Datei) bei den Veranstaltern eingereicht werden soll.
3. Führen Sie Ihr Programm im Rahmen einer max. 20-minütigen Präsentation in der letzten Semesterwoche am 31.1. bzw. 3.2.2005 vor.