

Lehrstuhl für Programmiersprachen  
und Übersetzerkonstruktion

Institut für Informatik  
Christian-Albrechts-Universität zu Kiel  
Prof. Dr. M. Hanus, S. Fischer



10. Übung zur Vorlesung „Logikprogrammierung“  
Sommersemester 2006

---

**Abgabe:** am Mittwoch, dem 28.6.2006 in der Vorlesung  
**29.6. und 6.7.: Übung am Wilhelm-Selig-Platz 3 in Raum 3!**

**Aufgabe 28** (Programmieraufgabe) 5 Punkte

Implementieren sie ein Prädikat `stars(Commands)`, das eine Liste von Kommandos als Parameter hat und Sterne entsprechend der Kommandosequenz auf dem Bildschirm ausgibt. Mögliche Kommandos sind `left`, `up`, `right`, `down` und `star`. Die ersten vier bewegen einen imaginären Cursor in die entsprechende Richtung und das letzte zeichnet einen Stern an die aktuelle Cursor-Position.

Der Benutzer soll an beliebiger Stelle anfangen dürfen zu zeichnen und das Bild soll immer oben links ausgerichtet sein. Sie müssen also das Koordinatensystem eventuell an die Kommandosequenz anpassen.

**Aufgabe 29** (Programmieraufgabe) 5 Punkte

Damit man sich an heißem Kaffee aus der Kaffeemaschine nicht verbrennt, ist es ratsam, den Kaffee vor dem Trinken einige Zeit abkühlen zu lassen. Problematisch ist jedoch die Frage, wie lange dieser Abkühlvorgang dauern sollte, insbesondere wenn man in Betracht zieht, zu einem beliebigen Zeitpunkt, dem Kaffee Milch hinzuzufügen. Der Abkühlvorgang kann ungefähr folgendermaßen mathematisch beschrieben werden:

$$T = (AnfangsTemp - RaumTemp) * \exp(Base, -1 * Const * Zeit) + RaumTemp$$

falls  $Zeit < AbkZeitp$  und

$$T = (StartTemp - RaumTemp) * \exp(Base, -1 * Const * RestZeit) + RaumTemp$$

sonst, wobei  $StartTemp = ((AnfangsTemp - RaumTemp) * \exp(Base, -1 * Const * AbkZeitp) + MilchTemp) / 2$ ,  $RestZeit = Zeit - AbkZeitp$ ,  $Const = 0.1$  und  $Base = 2$  sei.

Schreiben Sie ein Prädikat

`coffee(Temp, AnfangsTemp, RaumTemp, AbkZeitp, MilchTemp, Zeit)`

welches für Variablenbelegungen gemäß obiger Gleichungen erfüllt ist und testen Sie dieses Prädikat mit den Anfragen

- (a) `coffee(T, 80, 20, 0.5, 10, 1)`
- (b) `coffee(40, 80, 20, 0.5, 10, Z)`
- (c) `coffee(40, 80, 20, 0.5, 25, Z)`

**Hinweis:** Arithmetische Constraints können in SWI-Prolog nicht verwendet werden. In Sicstus muss zunächst der Befehl `use_module(library(clpr)).` eingegeben werden. Beachten Sie weiterhin, dass Constraint-Gleichungen und -Ungleichungen in Sicstus in geschweiften Klammern geschrieben werden müssen.