

## Softwaretechnik und Programmiersprachen I - Sommersemester 2011

### Übungsblatt 3

Besprechung der Aufgaben in 25.12.02.55 am Mittwoch 14:30-16:00 Uhr oder Freitag 10:30-12:00 Uhr  
Bei Fragen wenden Sie sich bitte an John Witulski: John.Witulski@uni-duesseldorf.de.

#### Aufgabe 3.1 (Lineare Resolution)

Sie kommen an einen Tatort. Dort sehen Sie ein eingeschlagenes Fenster, ein blutverschmiertes Beil und eine Leiche. Ein Verdächtiger wurde 100 Meter vom Haus entfernt festgenommen. Sie sind im Gerichtssaal und müssen den Richter beweisen, dass ein Mord stattgefunden hat.

Ein Mord hat stattgefunden, wenn eine Leiche, eine Einbruchsstelle, ein Mordwerkzeug und ein Verdächtiger gefunden wurden.

Schreiben Sie die oben genannten Aussagen in Prolog-Prädikaten und Fakten so um, dass Sie daraus einen Beweis mittels Linearer Resolution führen können und beweisen Sie damit, dass ein Mord stattgefunden hat.

#### Aufgabe 3.2 (Aussagenlogik und Resolution)

Formalisieren Sie die folgenden Aussagen in Aussagenlogik:

1. Das System heizt, wenn der Deckel geschlossen ist und Strom an ist.
2. Wenn das Lämpchen leuchtet, ist der Strom an.
3. Das Lämpchen leuchtet.
4. Der Deckel ist geschlossen.

Bringen Sie die Aussagen in die konjunktive Normalform und beweisen Sie mittels linearer Resolution, dass das System heizt.

#### Aufgabe 3.3 (SAT-Solver in Prolog)

Eine aussagenlogische Formel sei als Prolog-Term gegeben. Neben Konstanten und Variablen können Formeln mittels Und, Oder und Negation verknüpft werden. Für die Konstanten true und false werden die Terme `cst(true)` bzw. `cst(false)` und für die Verknüpfungen `and/2`, `or/2` und `not/1` verwendet.

Die aussagenlogische Formel

$$\neg(\text{true} \wedge \text{false}) \vee \text{false}$$

sähe als Prolog-Term also folgendermaßen aus:

$$\text{or}(\text{not}(\text{and}(\text{cst}(\text{true}), \text{cst}(\text{false}))), \text{cst}(\text{false}))$$

Schreiben Sie ein Prädikat `istrue/1`, das genau dann wahr ist, wenn die übergebene Formel wahr ist.

Zum Beispiel:

```
?- istrue( or(not(and(cst(true), cst(false))), cst(false)) ).
```

```
yes
```

```
?- istrue( or(not(cst(true)), cst(false)) ).
```

```
no
```

Falls in der Formel Variablen verwendet werden, sollen mögliche Lösungen gefunden werden:

```
?- istrue( or(not(and(cst(true), cst(A))), cst(B))).
```

```
A = false ? ;
```

```
B = true ? ;
```

```
no
```

Tipp: Verwenden sie für die Negation nicht die Prolog-Negation, sondern implementieren Sie (mit `fail`) ein zweites Prädikat `isfalse/1`, das angibt, ob eine Formel falsch ist.

```
?- fail.
```

```
no
```