

4. Übungsblatt zur Künstlichen Intelligenz

Wintersemester 2004/2005

Aufgabe 4.1 (Aussagenlogik: Erstellen von Formeln) Finden Sie eine Formel F , die die drei atomaren Formeln A, B und C enthält und folgende Eigenschaft ausweist: Für jede Interpretation $\mathcal{A} : \{A, B, C\} \rightarrow \{0, 1\}$ gilt, dass das Ändern irgendeiner der Werte $\mathcal{A}(A), \mathcal{A}(B), \mathcal{A}(C)$ auch $\mathcal{A}(F)$ ändert.

Aufgabe 4.2 (Aussagenlogik: Umwandlung von Ausdrücken)

1. Wandeln Sie den Ausdruck $(A \vee B) \wedge (A \vee C)$ in einen äquivalenten Ausdruck um, der *keine* Disjunktionen enthält.
2. Wandeln Sie den Ausdruck $((A \wedge B) \vee ((A \vee C) \wedge (B \vee D) \wedge \neg(C \vee D))) \wedge (A \vee D)$ in einen äquivalenten Ausdruck um, der keine Konjunktionen enthält.

Aufgabe 4.3 (Aussagenlogik: Erfüllbarkeit) Ist die unendliche Formelmeng

$$M = \{A_1 \vee A_2, \neg A_2 \vee \neg A_3, A_3 \vee A_4, \neg A_4 \vee \neg A_5, \dots\}$$

erfüllbar?

Aufgabe 4.4 (Boolesche Funktionen) Sei \diamond ein Symbol für die Boolesche Verknüpfung $x \diamond y = (\neg x \wedge y) \vee (x \wedge \neg y)$.

1. Wie wird die boolesche Funktion \diamond normalerweise genannt?
2. Leiten Sie die disjunktive Normalform für den Ausdruck $A = (x \diamond y) \diamond z$ her.
3. Ein Programmable Logic Array (PLA) aus der Technische Informatik ist ein Bausteingatter mit Logikgattern, dessen Logik vom Benutzer programmiert werden kann. Zeichnen Sie ein PLA, das die Ausdrücke $A = (x \diamond y) \diamond z$ und $B = x \diamond y$ realisiert.

Aufgabe 4.4 (Aussagenlogik: Resolution) Wenn das Einhorn ein Fabeltier ist, dann ist es unsterblich, aber wenn es kein Fabeltier ist, dann ist ein sterbliches Säugetier. Wenn das Einhorn entweder unsterblich oder ein Säugetier ist, dann hat es ein Horn. Das Einhorn ist märchenhaft, wenn es ein Horn besitzt.

1. Drücken Sie das obige Wissen mit Aussagenlogik aus.

2. Können Sie beweisen, dass das Einhorn ein Fabeltier ist? Falls ja, geben Sie einen Beweis an.
3. Können Sie beweisen, dass das Einhorn märchenhaft ist? Falls ja, geben Sie einen Beweis an.
4. Können Sie beweisen, dass das Einhorn ein Horn besitzt? Falls ja, geben Sie einen Beweis an.

Aufgabe 4.5 (Boolesche Algebra) Ein Missionar verfährt sich in Süd-Kalifornien auf der Suche nach Disneyland. An einer Weggabelung bleibt er stehen. Er fragt sich, ob er nach links oder rechts fahren soll. Er weiß außerdem, dass zwei Motorrad-Gangs die Gegend unsicher machen. Die Mitglieder der einen erzählen stets die Wahrheit, die der anderen lügen immer. Wenn jemand vorbeikommt, welche Frage soll der Missionar stellen?

Hilfe: Eine Frage zu stellen bedeutet hier, den Befragten zu bitten, eine Boolesche Funktion auszuwerten und das Ergebnis mitzuteilen. Auch der Lügner berechnet dann zuerst die ganze Wahrheit, bevor er sie invertiert.

1. Stellen Sie die Wahrheitstafel auf.
2. Geben Sie die disjunktive Normalform an.

Aufgabe 4.6 (Einarbeitung in Prolog)

In Aufgabe 1.4 sollten Sie Prolog (SWI-Prolog) auf ihren Rechnern installieren. Nach der Installation auf einem Unix System befindet sich der Prolog-Compiler in `/usr/local/bin/pl`.

Um das Prologsystem zu starten, müssen Sie in der unix Shell `/usr/local/bin/pl` tippen (Falls Suchpfade gesetzt sind, reicht `pl`). Sie werden dann den Prolog-Prompt `1 ?-` sehen. Um Prolog zu beenden, müssen Sie `halt.` eingeben.

Beispiel:

```
korfu /home/nuechter> pl
Welcome to SWI-Prolog (Multi-threaded, Version 5.4.2)
Copyright (c) 1990-2003 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.
```

For help, use `?- help(Topic).` or `?- apropos(Word).`

```
1 ?-
```

Nachdem Sie das Prologsystem gestartet haben, können Sie Fakten und Regeln direkt eingeben, indem Sie `consult(user).` tippen. Beispiel:

```
1 ?- consult(user).
```

Sie bekommen jetzt einen neuen Prompt `|:.` An dieser Stelle können Sie neue Fakten und Regeln eingeben, zum Beispiel:

```
|: man(socrates).
|: mortal(X) :- man(X).
|: end_of_file.
% user://1 compiled 0.00 sec, 672 bytes
```

```
Yes
2 ?-
```

In diesem Beispiel haben wir den Fakt, dass Sokrates ein Mensch ist und die Regel, dass alle Menschen sterblich sind, eingegeben. Alle Prädikatsnamen und Konstanten müssen mit Kleinbuchstaben beginnen. Variablen beginnen mit einem Großbuchstaben. Wenn Sie mit dem Eingeben der Fakten fertig sind, können Sie `end_of_file.` oder `^d` eintippen. Sie kehren dann zu dem ursprünglichen Prompt zurück und können nun Fragen bezüglich der eingegebenen Fakten stellen. Beispiel:

```
2 ?- man(socrates).
```

```
Yes
3 ?- mortal(X).
```

```
X = socrates <return>
```

```
Yes
4 ?-
```

Alternativ können Sie ein Semikolon eingeben (das „oder“ in Prolog). Dann versucht der Interpreter, ein weiteres X zu finden, das die Formel erfüllt. Wenn er ein solches nicht mehr findet, antwortet er mit „No“.

```
5 ?- mortal(X).
```

```
X = socrates ;
```

```
No
6 ?-
```

Falls Sie ihre Fakten und Regeln in einer Datei `test.pl` abgespeichert haben, können Sie die Datei mit dem Befehl `consult('test.pl')` laden. Eventuell auftretende Endlosschleifen lassen sich durch `^c` and anschließend `a` abbrechen.

Reproduzieren Sie das obige Beispiel mit Ihrem Prolog-System!