Aufgabe 1 (20 Punkte)

Dialogische Logik

a) Was ist eine formal wahre Behauptung? Welche Aussageschemata sind allgemeingültig?

b) Überprüfen Sie die klassische Gültigkeit der Implikation

$$((A \longrightarrow B) \wedge \neg B) \prec \neg(A \wedge B)$$

anhand einer Wahrheitswertetafel!

c) Zeigen Sie durch einen Hypothesendialog, dass $A \vee B, \neg A \prec B$ eine gültige Implikation ist!

Aufgabe 2 (15 Punkte)

a) Disjunktive Normalform (DNF)

i) Was ist die Disjunktive Normalform (DNF)? Worin unterscheidet sie sich zur Konjunktiven Normalform (KNF)?

ii) Wandeln Sie die Formel $(A \longrightarrow B) \wedge \neg(C \wedge D)$ in die DNF um!

b) Aussagenlogische Resolution: Zeigen Sie mit der Resolutionsmethode, dass

$$(\neg B \wedge \neg C \wedge D) \vee (\neg B \wedge \neg D) \vee (C \wedge D) \vee B$$

eine Tautologie ist.

Aufgabe 3 (10 Punkte)

Quantorenlogische Resolution

a) Bei quantorenlogischen Formeln sind Objektvariablen zu berücksichtigen. Wodurch muss das aussagenlogische Resolutionsverfahren ergänzt werden? Geben Sie für diese Ergänzung eine Definition an!

b) Geben Sie, falls existent, zur folgenden Menge den allgemeinsten Unifikator an! Variablen sind als Großbuchstaben notiert, Konstanten als Kleinbuchstaben.

$$\{f(Y, h(X, Z)), f(g(X, Z), h(Z, X))\}$$

Aufgabe 4 (20 Punkte)

Prolog

a) Was sind Horn-Klauseln?

b) Gegeben sei das folgende Prolog-Programm:

```
geheim(A, B):- xyz(A, A, B).
xyz([], C, [ m | C]).
xyz([D|Es], F, [D| G]) :- xyz(Es, F, G).
```

Was ergibt die folgende Anfrage?

```
?- geheim([1,2,3], X).
```

c) Kodieren Sie folgende Datenbasis als Prolog-Programm:

Ein $Tiger$ ist größer als ($greater/2$) ein $Hund$.
Ein $Tiger$ ist größer als eine $Henne$.
Ein $Hund$ ist größer als ein $Hase$.
Ein $Elch$ ist größer als ein $Hund$.

Die *greater*-Relation sei transitiv. Erweitern Sie das Programm um eine Prolog-Regel, die dieser Eigenschaft Rechnung trägt.

d) Implementieren Sie das Prädikat `member(Element, Liste)` in Prolog. Das Prädikat ist erfolgreich, wenn `Element` mit einem Element der `Liste` unifiziert werden kann. Verwenden Sie keine vordefinierten Prolog-Prädikate!

Beispiel eines Programmaufrufs:

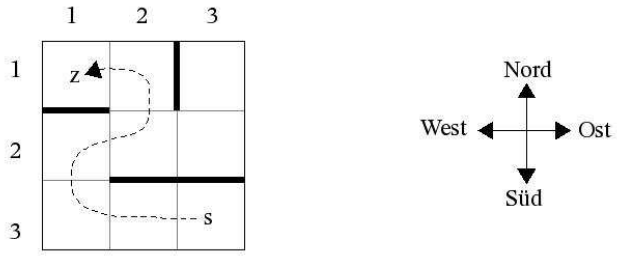
```
?- member(b, [a,b,c,d]).
yes
```

Aufgabe 5 (25 Punkte)

Prolog

a) In einem Labyrinth soll der Weg vom Startpunkt s zum Ziel z gefunden werden. Das Labyrinth besteht aus mehreren Feldern, die in einem Gitter angeordnet sind. Jedes Feld f hat einen eindeutigen Namen, welcher sich aus den x und y-Koordinaten zusammensetzt: $f_{x,y}$ bezeichnet das Feld in der x-ten Zeile und y-ten Spalte. Sind zwei benachbarte Felder nicht durch eine Wand getrennt, kann man von einem Feld in das andere gehen. Dabei bewegt man sich in einer der Himmelrichtungen „Nord“, „Süd“, „West“, „Ost“.

Der Weg vom Start zum Ziel soll als Liste von Richtungsanweisungen, die als Himmelsrichtungen anzugeben sind, ermittelt werden. (siehe auch Teilaufgabe ii))



i) Repräsentieren Sie das Labyrinth im Hinblick auf Teilaufgabe ii) als Prolog-Datenbasis.

ii) Schreiben Sie nun das Prolog-Programm `labyrinth(Start, Ziel, Weg)`, welches den Weg vom `Start` zum `Ziel` als Liste von Richtungsanweisungen (`Weg`) findet. Damit Ihr Programm terminiert, muss es eine Liste der schon besuchten Felder verwalten. Verwenden Sie außer dem `member/2`-Prädikat aus Aufgabe 4 d) keine vordefinierten Prolog-Prädikate!

Beispiel eines Programmaufrufs, welcher den gestrichelt eingezeichneten Weg für das obige Labyrinth findet:

```
?- Start = f33, Ziel = f11, labyrinth(Start, Ziel, Weg).
Weg = [west, west, nord, ost, nord, west] .
```

b) Schreiben Sie ein Prolog-Programm `nth(Ls, K, Result)`, das aus einer Liste `Ls` das `K`-te Element (Zählung beginnt bei 0) mit `Result` unifiziert. Verwenden Sie keine vordefinierten Prädikate. Beispiel eines Programmaufrufs:

```
?- nth([a, b, c ,d ,e ], 3, Result).
Result = d.
```