Fachhochschule Wedel
Prof. Dr. Uwe Schmidt

7. August 1997

Aufgaben zur Klausur **Grundlagen der Programmierung** und **Logische Programmierung** im SS 97 (WI03)

Zeit: 120 Minuten erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 7 Seiten

---

Aufgabe 1:

Zeigen Sie durch Transformation, daß die Aussage $(b \Rightarrow c) \Rightarrow (a \wedge b \Rightarrow a \wedge c)$ ein Satz der Aussagenlogik ist

$$(b \Rightarrow c) \Rightarrow (a \wedge b \Rightarrow a \wedge c)$$

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

Aufgabe 2:

Gegeben seien die Variablen

var f : array $[0..n-1]$ of Z;
var g : array $[0..m-1]$ of Z;
var $w1, w2$: Z;

Beschreiben Sie folgende Sachverhalte mit Hilfe der Prädikatenlogik

1. f enthält nur Werte, die auch in g vorkommen

   .....

   .....

   .....

2. f enthält alle Werte, die in g vorkommen

   .....

   .....

   .....

3. f enthält nur die Werte aus den Variablen $w1$ und $w2$

   .....

   .....

   .....

4. f enthält höchstens 2 verschiedene Werte

   .....

   .....

   .....

Aufgabe 3:

Transformieren Sie die folgende Funktion in eine gleichwertige Funktion, die mit einer Schleife arbeitet. Benutzen Sie hierzu Techniken aus der Vorlesung.

$qs(n : \mathsf{N}_0, b : \mathsf{N}_0) : \mathsf{N}_0$
  if $n = 0$
  then 0
  else $n \ \mathsf{mod}\ b + qs(n \ \mathsf{div}\ b, b)$

Die vollständig transformierte Funktion:

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

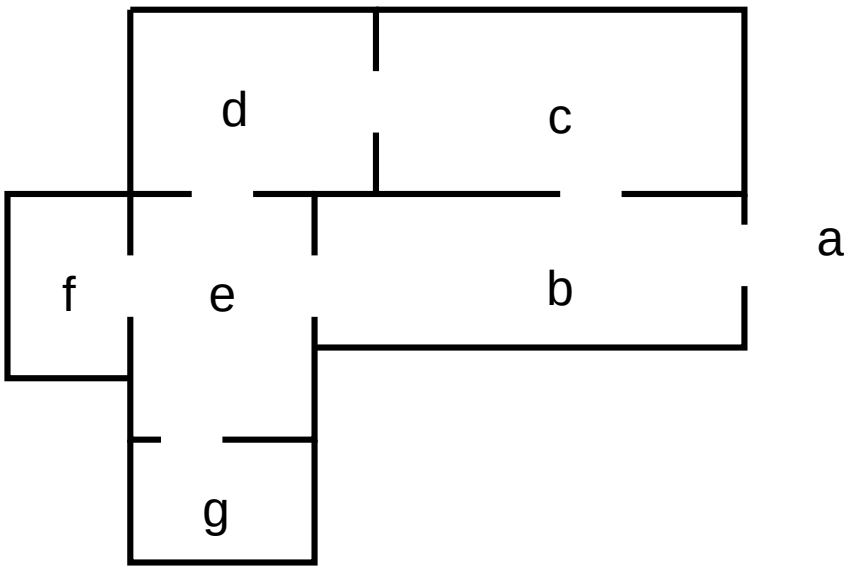
. . . . . .

. . . . . .

. . . . . .

Aufgabe 4:

In dieser Aufgabe sollen Prolog–Prädikate entwickelt werden, um Wege in einem Labyrinth zu finden. Die Grafik zeigt ein Labyrinth mit 6 Räumen, b bis g und der Außenwelt a. Die Räume sind durch Türen miteinander verbunden.



Dieses Labyrinth kann durch die folgenden Prolog–Fakten repräsentiert werden:

```
tuer(a,b).
tuer(b,e).
tuer(b,c).
tuer(d,e).
tuer(c,d).
tuer(e,f).
tuer(g,e).
```

Gesucht ist ein dreistelliges Prädikat gehe(+Start, +Ziel, +besuchteRaeume), das einen Weg sucht von einem Ausgangsstandpunkt Start zu einem Ziel, dabei aber höchstens einmal durch einen Raum führt. Der Parameter besuchteRaeume, eine Liste, dient zum Merken der schon besuchten Räume. Beachten Sie, daß die Türen in beide Richtungen durchschritten werden können.

Die Regeln für das Prädikat gehe(S, Z, L):

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Wie sieht eine Anfrage aus, ob man von außen zum Raum g kommen kann?

.....

Wie sieht eine Anfrage aus, ob man von c nach g kommen kann, ohne den Raum a zu betreten?

.....

Aufgabe 5:

Entwickeln Sie ein 3-stelliges Prädikat nextto(?X, ?Y, +L), das Erfolg hat, wenn X und Y benachbarte Elemente in der Liste L sind.

nextto soll also z.B. die folgenden Ergebnisse liefern:

```
?- nextto(3, 4, [1, 2, 3, 4, 1]).
yes
?- nextto(3, X, [1, 2, 3, 4]).
X = 4
?- nextto(X, 2, [1, 2, 3, 2, 1]).
X = 1;
X = 3;
no
```

Die Regeln für nextto(X, Y, L):

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Fachhochschule Wedel
Prof. Dr. Uwe Schmidt

7. August 1997

Aufgaben zur Klausur **Grundlagen der Programmierung** im SS 97 (II13)

Zeit: 120 Minuten erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 6 Seiten

---

Aufgabe 1:

Zeigen Sie durch Transformation, daß die Aussage $(b \Rightarrow c) \Rightarrow (a \wedge b \Rightarrow a \wedge c)$ ein Satz der Aussagenlogik ist

$$(b \Rightarrow c) \Rightarrow (a \wedge b \Rightarrow a \wedge c)$$

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

$\Leftrightarrow$ *Begründung :* .....

.....

Aufgabe 2:

Gegeben seien die Variablen

var f : array $[0..n-1]$ of Z;
var g : array $[0..m-1]$ of Z;
var $w1, w2$: Z;

Beschreiben Sie folgende Sachverhalte mit Hilfe der Prädikatenlogik

1. f enthält nur Werte, die auch in g vorkommen

   .....

   .....

   .....

2. f enthält alle Werte, die in g vorkommen

   .....

   .....

   .....

3. f enthält nur die Werte aus den Variablen $w1$ und $w2$

   .....

   .....

   .....

4. f enthält höchstens 2 verschiedene Werte

   .....

   .....

   .....

Aufgabe 3:

Transformieren Sie die folgende Funktion in eine gleichwertige Funktion, die mit einer Schleife arbeitet. Benutzen Sie hierzu Techniken aus der Vorlesung.

$qs(n : \mathsf{N}_0, b : \mathsf{N}_0) : \mathsf{N}_0$
  if $n = 0$
  then 0
  else $n \ \mathsf{mod}\ b + qs(n \ \mathsf{div}\ b, b)$

Die vollständig transformierte Funktion:

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Aufgabe 4:

Berechnen Sie zu den Anweisungen und Nachbedingungen mit Hilfe der Beweisregeln für Zuweisungen und Anweisungsfolgen die zugehörige Vorbedingungen und wenn möglich vereinfachen Sie diese. Alle Variablen seien Variablen für ganzzahlige Werte

1. $\{ V \}$ $\quad i, j := i - 3, j + 2 \quad$ $\{ i + j > 0 \}$

   .....

2. $\{ V \}$ $\quad i, j := j + i, i + j \quad$ $\{ i \neq j \}$

   .....

3. $\{ V \}$ $\quad i := j + i; j := j + i \quad$ $\{ i \neq j \}$

   .....

4. $\{ V \}$ $\quad i, j := i - 1, j - k \quad$ $\{ i * k - j = 1 \}$

   .....

5. $\{ V \}$ $\quad i, j := i \ \mathsf{div}\ i, j * 2 \quad$ $\{ i * j = k \wedge i \ \mathsf{mod}\ 2 = 0 \}$

   .....

Aufgabe 5:

Programmverfeinerung: Das folgende Programm berechnet für die Zahlenfolge $i^2 + 2 * i$ den n-ten Folgenwert in der Variablen y.

var $i, n, y, z : \mathsf{N}_0$;

{ true }
$i, y := 0, 0;$

{ $y = i^2 + 2 * i$ }
while $i \neq n$ do
    $i, y := i + 1, (i + 1)^2 + 2 * (i + 1)$

    { $y = i^2 + 2 * i$ }
end while

{ $y = i^2 + 2 * i \wedge i = n$ }

Verfeinern Sie dieses Programm so, daß es keine Potenzierung und keine Multiplikation mehr enthält. Die Zusicherungen brauchen nicht wiederholt werden. Es gibt eine Lösung, die bei Verwendung der Variablen z mit 3 Additionen auskommt.