Prof. Dr. Manfred Schmidt-Schauß
Künstliche Intelligenz/Softwaretechnologie

Fachbereich Informatik (20)
Johann Wolfgang Goethe-Universität Frankfurt am Main

Praktische Informatik 1:

Wintersemester 1997/98

Übungsblatt Nr. 5

Abgabe: Freitag 9. Januar 1998 vor der Vorlesung

Bonus-Aufgabe 1 (10 Punkte)

1. Notieren Sie die ersten 10 Schritte der Auswertung von `fib 6`, `quadrat (47 * 11)`, `quadrat (fib 12)` und `ifthenelse True (quadrat 8) (fib 800)` sowohl in normaler als auch in applikativer Reihenfolge unter Verwendung von β- und Case-Reduktion.
2. Bestimmen Sie die Anzahl aller möglichen Reduktionsfolgen zu `fakultaet 4` und `quadrat (quadrat 7)`, wobei `quadrat` und `fakultaet` definiert sind wie in der Vorlesung.

Bonus-Aufgabe 2 (5 Punkte)

Bestimmen Sie die genaue Anzahl der Normalordnungs-Reduktionen für `primzahlq n` in Abhängigkeit von `n` wenn `n` eine Primzahl ist. Wieviele `+`, `==`, `<` Operationen und wieviele β- und Case-Reduktionen müssen durchgeführt werden abhängig von `n`? Von welcher Größenordnung ist die Zahl der Reduktionen (O und Ω).

Bonus-Aufgabe 3 (5 Punkte)

Welche Variablen treten in den Unterausdrücken der folgenden Ausdrücke frei auf und welche treten gebunden auf?

1. `potenz_log 3 n`
2. `let z = x + 4 in quadrat z`
3. `(let z = 4 in z * 10) (let x = z in x + 3)`
4. `\ x -> x + 16`
5. `(\ x -> x + 16) 4`
6. `let x = 4 in x * (\ x -> x * x)`
7. `(let x = 4 in x * (\ x -> x * x)) 4`

Hinweis: Auch der ganze Ausdruck ist Unterausdruck seiner selbst.

Aufgabe 4 (10 Punkte)

Beweisen Sie *modus ponens* und $(A \Rightarrow B) \wedge (C \Rightarrow B) \Rightarrow (A \Rightarrow C)$ mit Hilfe von Wahrheitstabellen. Das *exklusive Oder* kann mit folgender Wahrheitstabelle definiert werden:

A	B	$A \oplus B$
T	T	F
T	F	T
F	T	T
F	F	F

beweisen Sie mit Hilfe von Wahrheitstabellen, daß $\oplus$ assoziativ und kommutativ ist.

Aufgabe 5 (10 Punkte)

Betrachten wir zunächst Aussageformen ohne Variablen. Zwei Möglichkeiten diese darzustellen sind die konjunktive und die disjunktive Normalform. Beide Normalformen können als Liste einer Liste von Wahrheitswerten in einem Programm verwendet werden. Programmieren Sie eine Funktion `interpretiere`, die als Argumente je eine Funktion für die Interpretation des inneren und des äußeren Junktors sowie eine solche Liste von Listen von Wahrheitswerten (`[[Bool]]`) verwendet, um den Wahrheitswert der Aussage zu bestimmen. Erlauben Sie "und", "oder", "exklusives oder" und "nicht-und" als Junktoren.

Testen Sie Ihr Programm z.B. mit

```
b1 = [[True, False]]
b2 = [[True, False], [True, False]]
b3 = [[True, False, True], [False]]
```

und verschiedenen Interpretationen für die Junktoren.

Aufgabe 6 (20 Punkte)

Erweitern Sie das Programm für Aufgabe 5. Verwenden Sie den Datentyp:

```
data Aussage
 = TV Bool
 | PV Bool Int
 deriving (Eq,Show)
```

In den `TV Bool` werden die Wahrheitwerte kodiert, in den `PV Bool Int` werden die aussagenlogischen Variablen (nur der Index) kodiert, das erste Argument soll dabei angeben, ob die Variable negiert auftritt oder nicht.

Programmieren Sie eine Funktion `vereinfache`, die ähnlich wie `interpretiere` arbeitet. Sehen Sie hier vor, daß "und", und "oder" die Junktoren in der äusseren oder den inneren Listen sein könnten, "exklusives oder" aber nur in den inneren Listen auftreten soll.

Hinweis: Möglicherweise hilft es Ihnen die Aufgabe in zwei Teilen zu bearbeiten: der eine führt die Vereinfachungen durch bei denen Wahrheitswerte beteiligt sind, der andere solche bei denen nur aussagenlogische Variablen beteiligt sind.

Testen Sie Ihr Programm z.B. mit

```
a1 = [[TV True, TV False]]
a2 = [[TV True, TV False], [TV True, TV False]]
a3 = [[TV True, TV False, TV True], [TV False]]
a4 = [[PV True 1]]
a5 = [[PV False 1]]
a6 = [[PV True 1, TV True]]
a7 = [[PV True 1, TV False]]
a8 = [[PV True 1, PV True 2]]
a9 = [[PV True 1], [PV True 2]]
a10 = [[PV False 1, PV True 2]]
a11 = [[PV True 1, PV False 1]]
a12 = [[PV True 1, PV True 2, TV False]]
a13 = [[TV True, TV True, TV True, TV False, PV True 1]]
a14 = [[PV False 1, TV False, PV False 1]]
a15 = [[TV False, PV True 1], [TV True, PV False 1]]
```

und verschiedenen Interpretationen für die Junktoren.

Bonus-Aufgabe 7 (5 Punkte)

Wenn Sie Eigenschaften des exklusiven Oder in Ihrem Programm verwenden, die nicht in der Vorlesung gezeigt wurden, beweisen Sie diese.

Aufgabe 8 (10 Punkte)

Verwenden Sie den Datentyp

```
type Substitution = Int -> Bool
```

und programmieren Sie Funktionen, die `Aussage` einen Wahrheitswert zuordnen und schließlich `[[Aussage]]`.

Testen Sie Ihr Programm mit den Aussagen oben und einigen Substitutionen.

Aufgabe 9 (10 Punkte)

Für diese Aufgabe seien die Elemente der inneren Listen durch "oder" verbunden und die der äußeren Liste durch "und" (konjunktive Normalform). Programmieren Sie eine Funktion `negiere`, die unter dieser Einschränkung `[[Aussage]]` negieren kann. Testen Sie Ihr Programm mit

```
negiere [[PV False 1, PV False 2], [PV False 3, PV False 4]
        ,[PV False 5, PV False 6]]
```

Wie lang können die Ausdrücke schlimmstenfalls werden?