Institut für Informatik
der Universität München
Prof. Dr. M. Hofmann
Dr. J. Johannsen, Dr. M. Lange

WS 03/04

4.11.03

Übungen zur Vorlesung
Informatik I

Blatt 7

Abgabe der Hausaufgaben spätestens am 15.12.03, 11:00 Uhr. Programmieraufgaben über

`http://miles.tcs.informatik.uni-muenchen.de/info1/abgabe.php`,

schriftliche Aufgaben auf Papier im Briefkasten in der Theresienstraße 39, 1. Stock. Notieren Sie Namen, Matrikelnummern und Ihre Übungsgruppe auf den Blättern. Bearbeitung in Gruppen zu max. 3 Personen ist zulässig. Besprechung der Aufgaben in den Übungen ab 22.12.03.

Schriftliche Aufgabe S-28: **6 Punkte**

Sei U die Umgebung, die durch Ausführung der folgenden Ocaml-Anweisungen entsteht.

```
let x = false ;;
let y = true ;;
let f x y = x ;;
let rec g x = g x ;;
```

Geben Sie zu jedem der folgenden Ocaml-Ausdrücke an, welche seiner Teilausdrücke ausgewertet werden, wenn der Gesamtausdruck in der Umgebung U ausgewertet wird.

a) `x && (not y) || (not x) || y ;;`

b) `if not (x=y) then x || y else x && y ;;`

c) `f x (g y) ;;`

Schriftliche Aufgabe S-29: **6 Punkte**

Es sei **lgz** die durch den untenstehenden Ocaml-Ausdruck berechnete Funktion, d.h. genauer:
lgz $= W^{\emptyset}(e)$ für den folgenden Ausdruck e:

```
let rec lgz n = if n=1 then 0
                else if n mod 2 = 1 then lgz(n+1)
                     else lgz(n/2)+1 ;;
```

Zeigen Sie durch Verwendung des Satzes von der partiellen Korrektheit aus der Vorlesung (Folie 151), dass für alle $n \in \mathbb{N}$ mit $n \geq 1$ gilt: falls **lgz**(n) definiert ist (d.h. wenn die Rekursion terminiert), so ist **lgz**$(n) = \lceil \log_2(n) \rceil$.

– bitte wenden –

Schriftliche Aufgabe S-30: **8 Punkte**

Werten Sie die folgenden Ausdrücke mit der in der Vorlesung vorgestellten operationalen Semantik für Ocaml aus. D.h. geben Sie jeweils einen Wert w an, so dass für einen Ausdruck e gilt: $\emptyset, e \to w$. Führen Sie die Berechnungen vollständig aus, d.h. geben Sie alle Nebenrechnungen an!

a) `3*(let x=4 in x+1) - if 2>0 then 10 else 5`

b)
```
let rec pow = fun (n,m) -> if m=1 then n else n*pow (n,m-1)
in pow (3,2)
```

Halten Sie sich an folgende Notation. Angenommen, Sie wollen $U_1, e_1 \to w_1$ zeigen, und dabei müssen Sie sowohl $U_2, e_2 \to w_2$ als auch $U_3, e_3 \to w_3$ als Nebenrechnungen ausführen. Desweiteren erfordert $U_2, e_2 \to w_2$ die Nebenrechnung $U_4, e_4 \to w_4$, usw. Notieren Sie dies wie folgt.

1.	$U_1, e_1 \to w_1$	folgt aus 2. und 3.
2.	$U_2, e_2 \to w_2$	folgt aus 4.
3.	$U_3, e_3 \to w_3$	$\dots$
4.	$U_4, e_4 \to w_4$	$\dots$
$\vdots$	$\vdots$	