Institut für Informatik
Arbeitsgruppe Grundlagen der Künstlichen Intelligenz
Prof. Dr. B. Nebel

Klausur:	**Probeklausur Informatik I**
Datum und Uhrzeit:	21.12.2010, 10:15 Uhr
Prüfungsdauer:	90 Minuten
Raum:	Hörsaal 101-00-036
Erlaubte Hilfsmittel:	Bücher, Notizen und ähnliches dürfen nicht verwendet werden. Dasselbe gilt für elektronische Geräte (Taschenrechner, Mobiltelefon, PDA, ...).
Prüfer:	PD Dr. J.-G. Smaus

Nachname:	.....
Vorname:	.....
Matrikelnummer:	.....
Fach:	.....
Studiengang:	☐ Bachelor ☐ Master ☐ Lehramt ☐ Sonstiges: .....
Unterschrift:	.....

Anmerkungen:

- Füllen Sie dieses Deckblatt vollständig aus.
- Zusätzliche Blätter sind nur einseitig zu beschreiben.
- Zusätzliche Blätter sind mit Namen und Matrikelnummer zu versehen.
- Für jede Aufgabe ist eine neue Seite/Bogen zu beginnen.
- Mobiltelefone müssen ausgeschaltet werden.

Prüfungsunfähigkeit:
Durch den Antritt dieser Prüfung erklären Sie sich für prüfungsfähig. Sollten Sie sich während der Prüfung nicht prüfungsfähig fühlen, können Sie aus gesundheitlichen Gründen auch während der Prüfung von dieser zurücktreten. Gemäß der Prüfungsordnungen sind Sie verpflichtet, die für den Rücktritt oder das Versäumnis geltend gemachten Gründe unverzüglich (innerhalb von 3 Tagen) dem Prüfungsamt durch ein Attest mit der Angabe der Symptome schriftlich anzuzeigen und glaubhaft zu machen. Weitere Informationen hierzu können auf den Internetseiten des Prüfungsamtes nachgelesen werden.

Aufgabe	max.	err.	Bem.	Aufgabe	max.	err.	Bem.
1	**5**			**8**	**8**		
2	**10**			**9**	**5**		
3	**10**			**10**	**5**		
4	**8**			**11**	**10**		
5	**8**			**12**	**7**		
6	**12**			**13**	**8**		
7	**4**			**Summe**	**100**		

Note:	.....
Klausur eingesehen am:	.....
Unterschrift des Prüfers:	.....

Name und Matrikelnummer:

Aufgabe 1 (5 Punkte) Algorithmen und Programme

Was ist der Unterschied zwischen einem Algorithmus und einem Programm?

Name und Matrikelnummer:

Aufgabe 2 (5+5 Punkte) LEXIKALISCHE BINDUNG

(a) Was ist die Ausgabe des folgenden Scheme-Programms?

```
(define x 1)
(define y 5)

((lambda (x y)
   (+ (* 2 x) y))
 y x)

((lambda (a b)
   (+ (* 2 x) y))
 y x)
```

(b) Erklären Sie kurz, wodurch sich die beiden Ausdrücke unterscheiden.

Name und Matrikelnummer:

Aufgabe 3 (5+5 Punkte) FALLUNTERSCHEIDUNGEN

(a) Schreiben Sie einen Scheme-Ausdruck, der den Absolutbetrag einer Zahl n berechnet.

(b) Erweitern Sie den Ausdruck zu einer Scheme-Prozedur, die den Absolutbetrag einer Zahl berechnet.

Name und Matrikelnummer:

Aufgabe 4 (5+3 Punkte) RECORD-DATENTYPEN

Ein Schokokeks in Scheme ist durch folgende Record-Definition definiert:

```
(define-record-procedures chocolate-cookie
  make-chocolate-cookie chocolate-cookie?
  (chocolate-cookie-chocolate
   chocolate-cookie-cookie))
```

(a) Ergänzen Sie das folgende Fragment einer Prozedur, die das Gewicht eines Schokokekses berechnet (wenn Sie davon ausgehen, dass ein Schokokeks exakt so viel wiegt wie seine beiden Anteile):

```
(define chocolate-cookie-weight
  (lambda (c)
    (
                                    )))
```

(b) Erklären Sie kurz, welcher Konstruktionsanleitung Sie gefolgt sind.

Aufgabe 5 (8 Punkte) LISTEN

Die Sorte der Zahlenlisten soll `zahlenliste` heißen (d. h., die Sorte der Elemente ist `number`). Um sie zu definieren, haben wir folgende Record-Definition für die Sorte `nll` der *nichtleeren* Zahlenlisten erstellt:

```
(define-record-procedures nll
  kons nichtleer?
  (kopf rumpf))
(: kons                                          )
(: nichtleer?                                    )
(: kopf                                          )
(: rumpf                                         )
```

Ergänzen Sie die unvollständigen Signaturen.

Name und Matrikelnummer:

Aufgabe 6 (12 Punkte) SIGNATUREN

Betrachten Sie folgendes Scheme-Programmstück, in dem für einige Bezeichner Signaturen und Definitionen angegeben werden:

```
(: list-1 (list number))
(define list-1 (cons "Banana" (cons 1 empty)))


(: list-2 (list number))
(define list-2 (cons 1 (cons 4 empty)))


(: list-3 (list string))
(define list-3 (cons "Apple" (cons "Banana" empty)))


(: list-4 (list (list string)))
(define list-4 (cons list-3 empty))


(: list-5 (list string))
(define list-5 list-4)


(: list-6 (list %value))
(define list-6 (cons 1 (cons "Banana" empty)))
```

Kennzeichnen Sie, an welchen Stellen Signaturverletzungen vorliegen, jeweils mit einer kurzen Erklärung. Kennzeichnen Sie Stellen, an denen *keine* Signaturverletzungen vorliegen, durch Abhaken.

Aufgabe 7 (4 Punkte) REKURSION

Was berechnet die Prozedur `spam`?

```
(: spam (natural -> natural))
(define spam
  (lambda (n)
    (if (zero? n)
        1
        (* n (spam (- n 1))))))
```

Name und Matrikelnummer:

Aufgabe 8 (4+4 Punkte) LOKALE VARIABLEN

(a) Was ist die Ausgabe des folgenden Scheme-Programms?

```
(define a 5)

(let ((a 1)
      (b (+ a 1)))
  b)

(let* ((a 1)
       (b (+ a 1)))
  b)
```

(b) Erklären Sie kurz, wodurch sich die beiden Ausdrücke unterscheiden.

Name und Matrikelnummer:

Aufgabe 9 (2+1+1+1 Punkte) EIGENSCHAFTEN VON RELATIONEN

Sei $A = \{1, 2, 3\}$ und $R = \{(1, 2), (2, 3), (1, 3)\} \subseteq A \times A$ eine binäre Relation über A.

(a) Geben Sie zu R die Umkehrrelation R^{-1} an.

(b) Ist R reflexiv?

(c) Ist R transitiv?

(d) Ist R symmetrisch?

Name und Matrikelnummer:

Aufgabe 10 (5 Punkte) TRANSITIVE HÜLLE

Sei $A = \{1, 2, 3, 4\}$. Geben Sie zur binären Relation $R = \{(1, 2), (2, 3), (3, 4)\} \subseteq A \times A$ die transitive Hülle an, d. h., die kleinste Relation, die R enthält und transitiv ist.

Name und Matrikelnummer:

Aufgabe 11 (10 Punkte) INDUKTION

Beweisen Sie mittels vollständiger Induktion, dass für alle $n \in \mathbb{N}$ gilt:

$$\sum_{i=1}^{n} \frac{i}{(i+1)!} = 1 - \frac{1}{(n+1)!}$$

Name und Matrikelnummer:

Aufgabe 12 (5+2 Punkte) TERMINIERUNG

(a) Wird folgende Prozedur auf der Eingabe 10 terminieren? (Verschiedene Antworten sind möglich, es kommt auf eine plausible Erklärung an. Achten Sie auf die Signatur!)

```
(: l (number -> number))
(define l
  (lambda (n)
    (if (zero? n)
        0
        (+ 1 (l (/ n 2))))))
```

(b) Korrigieren Sie das Programm derart, dass es $l : \mathbb{N} \to \mathbb{N}$ mit $l(n) = \lceil \log_2(n+1) \rceil$ berechnet.

Name und Matrikelnummer:

Aufgabe 13 (4+4 Punkte) PROGRAMMIERUNG HÖHERER ORDNUNG

(a) Die Prozedur zum Falten einer Liste sieht folgendermaßen aus:

```
(: list-fold (_ (_ _ -> _) (list %b) -> %a))
(define list-fold
  (lambda (e f l)
    (cond
     ((empty? l)
      e)
     ((cons? l)
      (f (first l) (list-fold e f (rest l)))))))
```

Ergänzen Sie die fehlenden Sorten (angedeutet durch _) in der Signatur.

(b) Man kann Listenfaltung verwenden, um die Konkatenation von Listen zu definieren:

```
(: append ((list %a) (list %a) -> (list %a)))
(define append
  (lambda (xs ys)
    (
                                          )))
```

Ergänzen Sie den fehlenden Rumpf der Definition.