Informatik 1 (Prof. Klaus Ostermann)

Gedächtisprotokoll

6. Februar 2018

Die Bearbeitungszeit betrug 70min, keine Hilfsmittel zugelassen!

1 Programmieraufgabe 1

1.1 Datendefinition

Schreibe eine Datendefinition für einen Beitrag in einem Forum, der die Anzahl der Likes, den Namen des Autors und den Text speichert. Außerdem soll er eine Liste aus Antworten zu diesem Beitrag enthalten. Die Antworten sind selbst wieder Beiträge. (3 Punkte)

1.2 Funktionsdefinition

Implementiere eine Funktion, die alle Likes zusammenrechnet, auch diejenigen, die an Antworten dieses Beitrags vergeben wurden. Du brauchst keine Tests und Aufgabenbeschreibung schreiben, die Signatur genügt.

2 Fragen

2.1

```
(cond [#true (+ 2 2)] [else 42])
```

Wie lautet das Programm nach Ausführung des ersten Auswertungsschritts?

- ◯ `[#true (+ 2 2)]`
- ◯ `(cond [#true 4] [else 42])`
- √ **(+ 2 2)**
- ◯ 4

2.2

```
(define-struct question (answer))
; answer is one of
; - Boolean
; - false
(define-struct questions (answer answer))
```

Wie viele Daten kann questions annehmen?

- √ **9**
- ◯ 4
- ◯ 6
- ◯ 12

2.3

Was ist Refactoring?

- ◯ Auswerten der Funktion mit Hilfe der Kongruenzregel
- ◯ Untersuchung des Programms auf die Größe der Teilmenge, die aus dem Datenuniversum genutzt wird
- ◯ Einbettung des Programms in eine S-Expression
- √ **Zusammenfassen ähnlicher Funktionen zu einer allgemeineren Funktion**

2.4

Was sind magic-numbers?

- ◯ Namen, denen durch Konstantendefinitionen ein Wert zugewiesen wurde.
- √ **Zahlenliterale, deren Bedeutung sich nicht direkt erschließt.**
- ◯ ...
- ◯ ...

2.5

Man möchte Nord,Süd, West und Ost definieren. Welchen Datentyp nutzt man?

- ◯ rekursiven Datentyp
- ◯ Intervalltyp
- ◯ Produkttyp
- √ **Summentyp**

2.6

Was steht laut Entwurfsrezept im Template eines Produkttypen?

- ◯ cond-Ausdruck
- √ **Feldselektoren**
- ◯ Prädikatsfunktion
- ◯ Konstruktor

2.7

Wann werden laut Entwurfsrezept die Tests geschrieben?

- ◯ Nach den Datendefinitionen
- ◯ Nach der Implementierung
- √ **Nach dem Funktionskopf**
- ◯ Nach dem Template

2.8

Was ist ein Wert?

- ◯ `(define (foo x)(+ x 2))`
- ◯ `(+ 2 2)`
- ◯ `(define y 5)`
- √ `"Ich liebe Klausuren"`

2.9

```
Ordne quote, quasiquote und unquote ihren Symbolen zu
A quote  B unquote C quasiquote
a ’   b ‘     c ,
```

- √ **A a, B c, C b**
- ◯ A b, B a, C b
- ◯ A c, B a, C b
- ◯ A b, B c, C a

2.10

Was ergibt

```
(`(+ ,(+ 1 2) ,(+ 3 4)))
```

?

◯ `(list '+ (list 3) (list 7))`

◯ `(list '+ (list 'unquote (list '+ 1 2)) (list 'unquote (list '+ 3 4)))`

◯ 10

√ `(list '+ 3 7)`

2.11

`Was ergibt die Auswertung von ("a" * 3) ?`

◯ `"aaa"`

√ `error: "a" is not a function`

◯ `error: * needs at least two arguments`

◯ `aaa`

2.12

Die folgenden zwei Fragen beziehen sich auf dieses Programm:

```
(define (f x)
   (cond [(empty? x)]
         [(cons? x) (g (rest x))]))
(define (g x)
    (cond [(empty? x) empty]
          [(cons? x) (cons (first x) (f (rest x)))]))
```

`Was ergibt (f (list #true 24 48 #false)) ?`

√ `(list 24 #false)`

◯ `(list #true 24 48 #false)`

◯ `(list #true 48)`

◯ `(list (list #true 24) (list 48 #false))`

2.13

Welche ist die allgemeinste Signatur für f?

- ◯ List-of-Number -> List-of-Number
- ◯ List-of-Number -> Number
- √ **[X] List-of-X -> List-of-X**
- ◯ [X] X -> X

2.14

Die folgenden 2 Fragen beziehen sich auf diese Programm:

```
(define (f x)
     (cond [(false? x) x]
           [else (+ x 1)]))
```

`Was ist (f (f 2)) ?`

- ◯ 2
- ◯ `#false`
- ◯ Typfehler
- √ **4**

2.15

In der Vorlesung wurde der Datentyp MaybeNumber eingeführt. Welches ist eine mögliche Signatur für f?

- √ **MaybeNumber -> MaybeNumber**
- ◯ MaybeNumber -> Number
- ◯ Number -> MaybeNumber
- ◯ Number -> Number

2.16

```
; a pair-of is a structure (make-pair-of X Y)
Welche Funktionen werden durch diese Strukturdefinition implementiert?
```

- ◯ make-pair, pair?, pair-X, pair-Y
- √ **keine**
- ◯ pair?, pair-of-X, pair-of-Y
- ◯ make-pair-of, pair-of?, pair-of-X, pair-of-Y

2.17

```
; list-of-starships asteroid -> Boolean
; determines whether one of the starships collides with the asteroid
(define (collision-1 starships asteroid) ...)

; list-of-starships list-of-asteroids -> Boolean
; determines whether one of the starships collides with any asteroid
(define (collision-2 starships asteroids)
     (cond [(collision-1 starships (first asteroids)) ...]
           [...]))
```

Was muss in die Funktion collision-2 eingefügt werden, damit die Funktion funktioniert?

◯ `(collision-1 (rest starships) asteroids)`

√ `(collision-1 starships (rest asteroids))`

◯ `(collision-2 (rest starships) asteroids)`

◯ `(collision-2 starships (rest asteroids))`

2.18

Eine gültige URL wird durch diese Grammatik beschrieben:

```
  <A> ::= http:// | https:// | http://www. | https://www. | www.
  <B> ::= google | amazon | marketplace
  <C> ::= co | uk | de
  <D> ::=   | home | uni | market
<URL> ::= <A> (<B>.)^+ (<C>.)^+ /<D>
```

Welches ist keine gültige URL?

◯ http://www.google.marketplace.de/

◯ www.google.co.uk/market

◯ https://amazon.co.uk.de/

√ **http://amazon.co.uk.de/market/home**

2.19

```
(define (f x y)
   (string-append x (number->string y)))
```

Was ist die allgemeine Signatur zu der Funktion?

◯ string number -> string

◯ [X Y] X Y -> X

√ **[X] X number -> X**

◯ number string -> number

2.20

```
(define FOO (+ BAR 1))
(define BAR 3)
(* FOO FOO))
Was gibt das Programm bei Ausführung aus?
```

◯ 9

◯ 16

◯ 10

√ **error: BAR is used before definition**

2.21

```
<E> ::= (<A> <B>) | ...
<B> ::= 1 | 2 | 3 | 4 | ()
Was muss bei <E> ergänzt werden, damit (1 ()), (1 (2 3)),
 ((2 (3 4)) ()) ausgedrückt werden können?
```

◯ (<B> <A>)

√ **<B>**

◯ ()

◯ <A>

2.22

Was ist ein gültiger Auswertungskontext?

◯ `(* 4 (+ 2 3) [ ] (+ 4 5))`

√ `(* 4 5 [ ] (+ 4 5))`

◯ `(* (+ 1 2)  5 [ ] (+ 4 5))`

◯ `(* 4 [ ] [ ] [ ]))`

2.23

Was besagt die Kong-Regel?

- ◯ Ein Programm wird von links nach rechts und von oben nach unten ausgewertet.
- √ **Falls e_1 äquivalent zu e_2 ist, dann auch (f e_1) und (f e_2).**
- ◯ Man darf eine Funktion durch eine andere ersetzen, wenn sich die Bedeutung des Programms dadurch nicht verändert.
- ◯ Konstantendefinitionen werden direkt ausgewertet bevor sie in die Umgebung geschrieben werden.

2.24

Was ist eine Umgebung?

- ◯ (define (f x) (+ x foo)
- ◯ (define x y)
- √ **(define y 5)**
- ◯ (define Blubb (+ 2 7))

2.25

Was ist syntaktischer Zucker?

- ◯
- √ **Funktionen die einfach durch andere Funktionen ausgedrückt werden können (z.B. and durch if)**
- ◯
- ◯

2.26

Was besagt DRY?

- ◯ Ausnutzen von copy and paste um redondante Sequenzen schneller zu programmieren
- √ **Ersetzen von Mustern im Programm durch Funktionen die dieses Muster generieren**
- ◯
- ◯

2.27

Wie kann folgender Funktionsbody äquivalent ersetzt werden?

```
; Boolean Boolean -> Boolean
; ...
(define (f b1 b2)
    (and b1 (or (not b2) (not #true))))
```

◯ (not b1)

√ **(and b1 (not b2))**

◯ (and b1 b2)

◯ (or b1 b2)

2.28

Frage 28

◯

√

◯

◯

2.29

Frgae 29

◯

√

◯

◯

2.30

Frage 30

◯

√

◯

◯