Gedächtnisprotokoll Informatik I Klausur für Medizintechniker

14.02.2022

Dozent: Klaus Ostermann

Anmerkung:

Dieses Protokoll ist nahezu Wortgleich mit der Klausur. Wie euch sicher auffällt ist es auch sehr ähnlich wie die anderen Protokolle. Es wiederholen sich also sehr viele Fragen und auch die Programmieraufgabe war sehr ähnlich (eigentlich waren nur die Namen der Felder des structs anders). Wir haben beim Erstellen auf Rechtschreibung und Syntax geachtet – auch wenn es manchmal nicht danach aussieht. Alle Angaben ohne Gewähr

Das hier ist die verkürzte medtech Klausur vom Ostermann. Bearbeitungszeit waren 60 Minuten

Fertigt nach Möglichkeit auch ein Gedächtnisprotokoll eurer Klausur an und ladet es ins Forum hoch. Tipp: Am Ende der Klausur könnt ihr die Fragen auf einen Zettel schreiben und diesen mit nach Hause nehmen. Das hat bei uns niemanden gejuckt.

Programmieraufgabe 10 Punkte

Eine Firma stellt Bauteile (engl.: part) her. Diese haben einen Namen (engl.: name), festgelegte Kosten (engl.: cost) und bestehen eventuell aus einem oder mehreren Komponenten (engl.: components) die wiederum Bauteile sind, es sind also endlose Verschachtelungen möglich. Für diese Aufgabe dürfen Sie alle Ihnen aus der Vorlesung bekannten Konzepte verwenden, insbesondere List-of-X

1.1

Schreiben Sie eine Datendefinition für part mit seinem Namen, seinen Kosten und einer Liste seiner Bestandteile. Nehmen Sie eventuell notwendige Daten- und Strukturdefinitionen selbst vor. Schreiben Sie ein Beispiel für ein Bauteil mit mindestens 2 Komponenten. (3 Pkt)

1.2

Schreiben Sie eine Funktion `total-cost`, die die Kosten eines Bauteils sowie die Kosten seiner zugehörigen Komponenten summiert. Tipp: Schreiben Sie eine Funktion, die die Kosten eines Bauteils berechnet, und eine zweite Funktion, die die Kosten einer Liste von Bauteilen berechnet. (7 Pkt)

Multiple Choice

1. Wie wird der folgende Ausdruck ausgewertet?
   `(and (e-1 e-2))`?
   a. `e-2` wird nur ausgewertet, wenn `e-1 #true` ergibt
   b. beide Argumente werden in jedem Fall nacheinander ausgewertet
   c. `e-2` wird nur ausgewertet, wenn `e-1 #false` ergibt
   d. `e-1` wird nur ausgewertet, wenn `e-2 #true` ergibt

2. Was ergibt die Auswertung des Ausdrucks
   `` `(+ 2 ,(+ 3 4) 5))`?
   a. `(list '+ (list 2 (list 7 (list 5))))`
   b. `(list '+ 2 (list '(+ 3 4)) 5`
   c. `(list '+ 2 7 5)`
   d. `14`

3. Welchen Datentyp wählen Sie wenn Sie einen Spielercharakter mit einem Level, einem Namen und Werten für die Attribute Stärke, Geschicklichtkeit, Konstitution, Weisheit, Intelligenz und Ausdauer speichern wollen?
   a. Rekursiver Datentyp
   b. Summentyp
   c. Produkttyp
   d. Primitiver Datentyp

4. Wie kann der folgende Ausdruck reduziert werden?
```
(define(f b1 b2)
 (or (and (not b1)
          (not (not #true))
          (b2)))
```
   a. `(or b1 b2)`
   b. `(b2)`
   c. `(or (not b1) b2)`
   d. `(or (and (not b1) #false) b2)`

5. Welche Ausgabe erzeugt der folgende Ausdruck?
```
(define  x (+ y 3))
(define y 5)
(define f (* x y))
```
   a. `40`
   b. `8`
   c. `5`
   d. Den Fehler `„y is used before its definition“`

6. Welche Ausgabe produziert
   `(+ 1 „1“)`?
   a. `2`
   b. `11`
   c. `„11“`
   d. Den Fehler `„function expects a number was given a string“`

Programm A:

```
(define (myfun x)

 (cond [(> x 10) (- x 10)]
       [else #false]))
```

7. Was ist die Signatur von Programm A?
   a. `MaybeNumber -> MaybeNumber`
   b. `Number -> Number`
   c. `MaybeNumber -> Number`
   d. `Number -> MaybeNumber`

8. Welche Ausgabe erzeugt `(myfun (myfun 2))`?
   a. `#false`
   b. `-8`
   c. Den Fehler: „`Function expects a Number was given a boolean`“
   d. `8`

9. Was ist eine Wichtige Eigenschaft oder Aufgabe von Tests?
   a. Tests können als Teil der Spezifikation einer Funktion verwendet werden.
   b. Tests können die Akkumulator-Invariante sicherstellen.
   c. Wenn alle Tests wie erwartet ablaufen, ist das Programm fehlerfrei.
   d. Tests können garantieren, dass keine Fehler beim Pattern Matching auftreten.

10. Was besagt die KONG Regel?
    a. `(cons 2 emtpy)` kann direkt zu 2 reduziert werden
    b. Wenn `e1` zu `e2` reduziert werden kann dann kann auch `(cons e1 empty)` zu `(cons e2 empty)` reduziert werden
    c. `(cond [#true e1] [else e2])` kann zu `e1` reduziert werden
    d. Wenn die Umgebung die Definition `(define (f x) (cons x empty))` enthält darf `(f 2)` zu `(cons 2 empty)` reduziert werden.

11. Welche Ausgabe produziert `(#true + 3)`?
    a. `#true`
    b. Den Fehler „`function expects a number was given a boolean`“
    c. Den Fehler „`open parenthesis expects a functioncall was given a #true`“
    d. Den Fehler „`function expects a boolean was given a number`“

12. Was ist ein Wert in BSL?
    a. `#true`
    b. `(define y 3)`
    c. `(string-append „ich liebe“ „klausuren“)`
    d. `(+ 5 2)`

13. Welche Ausgabe erzeugt
```
(not (or (not (= 3 5))(+40 2)))?
```
   a. `42`
   b. `#true`
   c. `#false`
   d. Den Fehler „`or expects a boolean was given 42`"

14. Was ist ein Vorteil des Templates für Summentypen? Das Template stellt sicher, dass man…
   a. alle Alternativen des Summentyps durch Funktionen erzeugen kann.
   b. alle Alternativen des Summentyps im Funktionskörper behandelt
   c. alle Alternativen des Summentyps durch Tests abdeckt.
   d. alle Alternativen des Summentyps durch Prädikate voneinander unterscheiden kann.

15. Was besagt das DRY Prinzip?
   a. Wenn Code in einem Programm doppelt vorkommt, besteht die Gefahr, dass die Kopien nicht das gleiche berechnen wie die Originale.
   b. Programmierer sollten niemals mit Copy & Paste arbeiten.
   c. Ausnutzen von copy and paste um redundante Sequenzen schneller zu programmieren
   d. Ersetzen von Mustern im Programm durch Funktionen, die dieses Muster generieren

16. Wozu wird das folgende in einem Schritt ausgewertet?
```
(cond [#true (+ 2 2)]
      [(> 1 0)(+ 1 1)])
```
   a. `4`
   b. `(+ 2 2)`
   c. `(cond [(> 1 0) (+1 1)])`
   d. `#true`

17. Was ist da Problem an folgendem Programm?
```
Number -> Number
(define (pred x)
 (cond [(> x 0) (- x 1)]
       [(= x 0) 0]
       [(< x 0) #false]))
```
   a. `cond` hat keinen `else` fall
   b. Beim Refactoring muss das `cond` als verschachtelte `if` ausdrücke beschrieben werden
   c. Die Funktion behandelt nicht den Fall, dass das übergebene Argument keine Zahl ist
   d. Der Rückgabewert `#false` ist Aufgrund der Signatur nicht erlaubt

18. Geben ist die Definition eines Auswertungskontexts für BSL mittels kontextfreier Grammatik aus dem Skript:
```
<E>::= [ ]
|((<name> <v> * <E> <e>*)
|(cond [<E> <e> ] {(<e> <e>)}*)
```
   Was ist ein gültiger Auswertungkontext?
   a. `(cond [] 2)`
   b. `(cond [#false []] (+ 40 2))`
   c. `(cond [#false x] (+ 40 []))`
   d. `(cond [#false x] (+[] 2))`

19. Was ist syntaktischer Zucker?
    Syntaktischer Zucker sind die Konzepte einer Programmiersprache…
    a. …die für die Programmausführung unwesentlich sind (z.B. Kommentare). Sie müssen vor dem parsen zu speziellen Ignore-Token entzuckert werden.
    b. …die nicht direkt einem Prozessorbefehl entsprechen (z.B. Funktionsdefinitionen). Sie müssen vor der Ausführung zu einer Sequenz von Maschinenbefehlen entzuckert werden.
    c. . … die von einem Editor zusätzlich zum Programm angezeigt werden (z.B. farbliche Markierung nach der Syntaxprüfung). Sie müssen vor dem Ausführen wieder zum eigentlichen Programm entzuckert werden
    d. …die leicht durch andere Konzepte ausgedrückt werden können (z.B. if durch cond). Sie müssen vor dem Ausführen in die Kernsprache entzuckert werden.

Programm B:

```
(define (f x)

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

20. Was ergibt die Auswertung von `(f (list #true 24 48 72 #false))`?
    a. `(list #true 48 #false)`
    b. `(list #true 24 48 72 #false))`
    c. `(list 24 72)`
    d. `(list #true 72)`

21. Was ist die allgemeinste Signatur von f?
    a. `[X] List-of-X -> List-of-X`
    b. `[XY] List-of-X -> List-of-Y`
    c. `List-of-Number -> List-of-Number`
    d. `List-of-Number -> Number`

22. Was sind „magic numbers"?
    a. Namen, denen durch Konstantendenitionen ein Wert zugewiesen wurde.
    b. Zahlenliterale, deren Bedeutung sich nicht direkt erschließt
    c. Zahlen die in der Umgebung des Programms verfügbar sind
    d. Zahlen, die dieselbe Struktur aufweisen wie das Programm selbst

23. Welchen Datentyp wählen Sie, wenn Sie die Charakterklasse (Krieger, Bogenschütze oder Heiler) eines Videospielcharakters darstellen sollen?
    a. Rekursiver Datentyp
    b. Summentyp
    c. Produkttyp
    d. Primitiver Datentyp

Gegeben ist die folgende Strukturdefinition:

```
(define-struct collision (top right bottom left))
```

24. Wie viele Werte umfasst der Teil des Datenuniversums den collision beschreibt?
    a. 4
    b. 2
    c. 16
    d. Unendlich viele

25. Was beschreibt `(define-struct collision(…))`
    a. Eine Sturkurdefinition für collision
    b. Produkttyp für collision
    c. Keines der genannten
    d. Strukturtyp für collision

26. Was ist eine gültige Instanz von `collision`?
    a. `(make-collision #true 42 #false "world")`
    b. `(make-collision #true #false #true #true)`
    c. `(make-collision #false #true #false)`
    d. `(make-collision #true (make-collision #true #false #false #false) #true #false)`

27. Um Welchen Datentyp handelt es sich bei `collision`?
    a. Rekursiver Datentyp
    b. Summentyp
    c. Produkttyp
    d. Primitiver Datentyp

28. Wann wird laut dem allgemeinen Entwurfsrezept die Funktionssignatur geschrieben?
    a. Vor den Datenbeispielen für Datendefinitionen
    b. Vor dem Funktionskopf
    c. Nach dem Funktionskopf
    d. Nach der Implementierung

29. Was ist eine Umgebung?
    a. (define (f x)(+ x 2)
       (define y x)
    b. (define (f x)(+ x 2)
       (define y (f 2))
    c. (define (f x)(+ x 2)
       (define y (+ 2 3))
    d. (define (f x)(+ x 2)
       (define y 5)

30. Eine gültige URL wird durch diese Grammatik beschrieben

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

Welches ist laut dieser (fantasie-) Grammatik keine gültige URL?

a. http://-http://www.google.co.uk.de/home
b. http://amazon.de/
c. www..google.co.uk/home
d. http://www..www.marketplace.amazon.de/market

Lösungsvorschlag

1 a
2 c
3 c
4 c
5 d
6 d
7 d
8 c
9 a
0 unsicher
1 c
2 a
3 unsicher b oder d
4 b
5 d
6 b
7 d
8 unsicher a
9 d
0 c
1 a
2 a
3 b
4 c
5 a
6 b
7 c
8 b
9 d
0 c