

Tutorium - Informatik 1

—

Montags 16 st.

Jules Kreuer, VL bei Prof. Grust



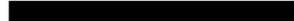
SDMG/Dettenmeyer - 2020 aus Stuttgarter Zeitung



SZ/BZ - 04.12.2020 <https://youtu.be/xQ22dpWrC6o>

Informatik 1

Modul INF1110 - für B.Sc *-Informatik, Physik und MedTech



- 🕒 Montag: 16 st.
- 📍 <https://zoom.us/j/91881562169>
- ✉ jules.kreuer@student.uni-tuebingen.de

i Das Zoom-Passwort gibts im Forum und auf Discord!

In diesem Tutorium zur Vorlesung Informatik 1, gehalten von Torsten Grust besprechen wir die Übungsbätter, bearbeiten gemeinsam Übungsaufgaben und wiederholen die wichtigsten Teile der Vorlesung. Das offizielle Forum zur Vorlesung finden Sie hier. Ein (offizielles) Skript zur Vorlesung gibt es nicht, jedoch findet sich eines in der Sammlung der Fachschaft Informatik auf GitHub.

Feedback zum Tutorium kann über das Feedbacksystem abgegeben werden. Vergangene Lehrevaluationen sind unter Evaluationen zu finden.

T1 - Einführung

**T2 - Einfache
Funktionen**

**T3 - Sig/Cond
/Reduktion**

Feedback

<https://juleskreuer.eu/feedback>

Feedback

Hier habt ihr die Möglichkeit (anonym) Feedback zu den Tutorien abzugeben. Ich werde versuchen euer Feedback umzusetzen!

Schreibt also ob ich meinen Job gut mache oder ob es Dinge gibt, die man verbessern kann.

i Bleibt bitte konstruktiv!

Fragen? Wünsche? Positives? Negatives?

Vorlesung:

Pseudonym:

ÜB 3

Reduktion und Vereinfachung

Informatik 1

Forum: <https://forum-db.informatik.uni-tuebingen.de/c/ws2021-info1>

Übungsblatt 3 (25.11.2020)

Abgabe bis: Mittwoch, 2.12.2020, 14:00 Uhr



Relevante Videos: bis einschließlich Informatik 1 - Chapter 03 - Video #015.

<https://tinyurl.com/Informatik1-WS2021>

Sprachebene „Die Macht der Abstraktion — Anfänger“

Aufgabe 1: [12 Punkte]

Reduziert die unten angegebenen Scheme-Ausdrücke (a) bis (c) nach den Reduktionsregeln [·] für Scheme, die in der Vorlesung besprochen wurden.

Wichtig: Notiert *alle* Reduktionsschritte genau wie in den Beispielen der Vorlesung zu *Chapter 3* auf den *Folien 5 bis 7* (nutzt dazu Racket-Kommentare).

- (a) Reduziert zunächst den **arithmetischen Ausdruck**

```
(* 2 (+ 20 1))
```

Reduziert anschließend den semantisch äquivalenten Ausdruck

```
(+ (+ 20 1) (+ 20 1))
```

Vergleicht die Anzahl der jeweils benötigten Reduktionsschritte.

- (b) Gegeben die folgenden **Konstanten- und Funktionsdefinitionen:**

```
(define pi 3.1415)  
(define perimeter  
  (lambda (r)
```

Schwierigkeiten bei ÜB 3

- Genau lesen :)
- Aufgabe 1:
 - Stur die Red-Regeln befolgen
- Aufgabe 2:
 - Tautologien beachten

2(b || !b) - Hamlet 3. Akt, 1 Satz

(if (>= y 11)  (if [#t / #f]
#t
#f) #t
#f)

 (if a
a
a)

2(b || !b) - Hamlet 3. Akt, 1 Satz

(if (>= y 11) → (if [#t / #f]
#t
#f)
#t

(if a
a
a)



Persian Cat Room Guardian - AnyaBoz 2013

2(b || !b) - Hamlet 3. Akt, 1 Satz

(if (\geq y 11) \longrightarrow (if [#t / #f]
#t
#f) #t
#f)

(if a \longrightarrow a bzw. (\geq y 11)
a
a)

2(b || !b) - Hamlet 3. Akt, 1 Satz

```
(one-of #t #f)    -> boolean
(boolean=? b #t) -> b
(boolean=? b #f) -> (not b)
(true? b)        -> b
(false? b)       -> (not b)
(if b #t #f)     -> b
(if b #f #t)     -> (not b)

(or b (not b))  -> #t
(and b (not b)) -> #f

(if (< y 10)    -> (< y 10)
    #t
    #f)
```

Schwierigkeiten bei ÜB 3

- Genau lesen :)
- Aufgabe 1:
 - Stur die Red-Regeln befolgen
- Aufgabe 2/3:
 - Tautologien beachten
- Aufgabe 4:
 - Abbruchbedingung der Reduktion anschauen
 - (heiner-or #t (/ 1 0)) vs. (or #t (/ 1 0))

Records

Was ist das?

Records

- Struktur die Werte trägt

Record <name>

- value-0 : type
- value-i : type
- value-n : type

- Funktionen für diese Struktur

- + make-<name>
- + <name>?
- + <name>-label-i>

Records

```
(define-record-procedures <name>  
  make-<name>           ; Konstruktor  
  <name>?               ; Prädikat  
  (<name>-<label-0>     ; Parameterliste  
   ...  
  <name>-<label-n>))
```

Records

```
(define-record-procedures person
  make-person           ; Konstruktor
  person?              ; Prädikat
  (person-name        ; Parameterliste
   person-height))
```

```
(define tutor (make-person 'Jules' 1.83))
(person-name tutor) → 'Jules'
```

Records Signaturen

Konstruktor

(: `make`-`<name>` (<type-0> ... <type-0> -> <name>))

Prädikat

(: <name>? (any-> `boolean`))

“Getter”

(: <name>-<label-i>? (<name> -> <type-i>))

Für JEDES Label eine Signatur!

ÜB 4

Records & BlackBox Testing

Übungsblatt 4 (2.12.2020)

Abgabe bis: Mittwoch, 9.12.2020, 14:00 Uhr



Relevante Videos: bis einschließlich Informatik 1 - Chapter 04 - Video #017.

<https://tinyurl.com/Informatik1-WS2021>

Sprachebene „Die Macht der Abstraktion — Anfänger“

Fehlerbehandlung in euren Abgaben

In diesem Übungsblatt 4 könnt ihr davon ausgehen, dass die Argumente für eure Funktionen nicht zu Fehlern (wie z.B. Division durch 0, ...) führen werden.

Aufgabe 1: [10 Punkte] Überprüft Kalenderdaten auf ihre Gültigkeit:

- Schreibt eine Daten- und eine Record-Definition für *Kalenderdaten*, welche sich aus der rein numerischen Repräsentation eines Datums (Tag, Monat, Jahr) zusammensetzt.
- Schreibt ein Prädikat `calendar-date-ok?`, das feststellt, ob ein Kalenderdatum-Record einem tatsächlichen Kalenderdatum entspricht, also korrekte Daten wie 1.1.1970 von unsinnigen wie 31.17.2016 unterscheidet. Schaltjahre sollen dabei ignoriert werden.
- Schreibt ein Prädikat `calendar-date-ok/leap-year?` das sich wie `calendar-date-ok?` verhält und zusätzlich Schaltjahre¹ berücksichtigt!

Hinweis: Zur Lösung der Aufgabe kann die eingebaute Funktion

```
(: modulo (integer integer -> integer))
```

hilfreich sein. Sie bestimmt den Rest einer ganzzahligen Division.

Aufgabe 2: [20 Punkte] Wir betrachten einen Garten mit Blumen und Rasensprenger:

- Eine *Blume* (`flower`) ist charakterisiert durch einen *Namen*, einen ganzzahligen *Wasserbestand* zwischen 0 und 100 sowie eine *Position* auf einem zweidimensionalen Gitter, das den Garten repräsentiert.
- Ein *Rasensprenger* (`sprinkler`) zeichnet sich durch seine *Reichweite* (gemessen in Metern) und *Effektivität* (ein reeller Wert zwischen 0 und 1) aus.

Feedback

<https://juleskreuer.eu/feedback>

Feedback

Hier habt ihr die Möglichkeit (anonym) Feedback zu den Tutorien abzugeben. Ich werde versuchen euer Feedback umzusetzen!

Schreibt also ob ich meinen Job gut mache oder ob es Dinge gibt, die man verbessern kann.

i Bleibt bitte konstruktiv!

Fragen? Wünsche? Positives? Negatives?

Vorlesung:

Pseudonym: