

Präsenzübungen zur Vorlesung
Deklarative Programmierung: Sommersemester 2018
Nr. 8

Hinweis: Ändern Sie den Sprachlevel der DrRacket-Umgebung auf „Die Sprache Racket“. Ändern Sie die erste Zeile Ihres Programmes auf „`#lang racket`“ bzw „`#lang typed/racket`“. Fügen Sie weiterhin „`(require 2htdp/abstraction)`“ zu Ihrem Programm hinzu.

Aufgabe 8.1: Coole Typen

- a) Erläutern Sie den Unterschied zwischen ungetypten, dynamisch getypten und statisch getypten Sprachen!
- b) Diskutieren Sie die Vor- und Nachteile von dynamischer/statischer Typisierung.
- c) Was bewirkt das Zuordnen eines *Contracts* zu einer Funktion?
- d) Formulieren Sie eine *provide*-Klausel, die der folgenden Funktion einen passenden Contract zuordnet.

```
; Number (list-of Number) -> (list-of Number)
; Adds the given number to all the elements in the given list
(define (add/all n lon) (...))
```

- e) Geben Sie die Typed Racket Signatur dieser Funktion an!
- f) Finden Sie ein Beispiel für ein Programm, das in BSL zulässig ist, in Typed Racket jedoch nicht.

Aufgabe 8.2: Advanced Dicing Technologies

Betrachten Sie die folgende Definition eines binären Suchbaums:

```
; A BST is one of:
; - (make-leaf Number)
; - (make-node Number BST BST)
; interp. as binary search tree for numbers

; Number BST -> BST
; Inserts the given number into the tree
(define (tree/insert value tree) (...))

; Number BST -> Boolean
; Checks if the given tree contains the given Number
(define (tree/contains value tree) (...))
```

- a) Implementieren Sie die Funktionen mithilfe von Strukturdefinitionen.
- b) Implementieren Sie die Funktionalität mithilfe des `define-type` Konstrukts.

Aufgabe 8.3: Wonder Tree



```
male(uranus) .
male(cronus) . father(uranus, cronus) . mother(gaia, cronus) .
male(zeus) . father(cronus, zeus) . mother(rhea, zeus) .
male(poseidon) . father(cronus, poseidon) . mother(rhea, poseidon) .
male(hades) . father(cronus, hades) . mother(rhea, hades) .
male(ares) . father(zeus, ares) . mother(hera, ares) .
male(hephaestus) . father(zeus, hephaestus) .
  ↳ mother(hera, hephaestus) .
```

```
female(gaia) .
female(rhea) . father(uranus, rhea) .
female(hera) . father(cronus, hera) . mother(rhea, hera) .
female(hestia) . father(cronus, hestia) . mother(rhea, hestia) .
female(demeter) . father(cronus, demeter) . mother(rhea, demeter) .
female(hippolyta) . father(zeus, hippolyta) .
female(athena) . father(zeus, athena) .
female(diana) . father(zeus, diana) . mother(hippolyta, diana) .
```

```
parent(P, C) :- father(P, C) .
parent(M, C) :- mother(M, C) .
son(S, P) :- parent(P, S), male(S) .
daughter(D, P) :- parent(P, D), female(D) .
```

```
grandfather(G, C) :- male(G), parent(G, P), parent(P, C) .
grandmother(G, C) :- female(G), parent(G, P), parent(P, C) .
```

Beantworten Sie folgende Anfragen durch Anwendung der Deduktionsregeln:

- a) Ist Gaia die Mutter von Cronus? (`?- mother(gaia, cronus) .`) 
- b) Ist Zeus ein Elternteil von Athena? (`?- parent(zeus, athena) .`) 
- c) Ist Zeus Großvater von Diana? (`?- grandfather(zeus, diana) .`) 