

Formale Techniken der Software-Entwicklung
Übungsblatt 1
Besprechung am 24.04.2015

Aufgabe 1:

Zeigen Sie, dass die folgenden aussagenlogischen Formeln Tautologien sind.

- (a) $A \implies (B \implies (A \wedge B))$
- (b) $(A \implies B) \implies ((A \implies \neg B) \implies \neg A)$

Aufgabe 2:

Zeigen Sie, dass die folgenden aussagenlogischen Formeln **keine** Tautologien sind. Sind sie erfüllbar? Falls ja, geben Sie jeweils eine Belegung an, die die Formel erfüllt.

- (a) $(A \implies C) \implies ((B \implies D) \implies ((A \vee B) \implies C))$
- (b) $(A \implies B) \implies ((B \implies \neg C) \implies \neg A)$

Aufgabe 3:

Zeichnen Sie die Struktur der folgenden binären Bäume und geben Sie jeweils eine Liste der Blätter an.

- (a) $((a \ b \ . \ c) \ . \ d)$
- (b) $((a \ b \ c) \ 1 \ 2 \ (3 \ . \ 4) \ 5)$

Aufgabe 4:

Definieren Sie eine Funktion (`pascal i j`), die den Binomialkoeffizienten $\binom{i}{j}$ berechnet, indem sie den j -ten Eintrag in der i -ten Zeile des in *Pascalschen Dreiecks* zurückgibt (siehe Abbildung 1).

				1					
				1	1				
			1	2	1				
		1	3	3	1				
	1	4	6	4	1				
	1	5	10	10	5	1			
1	6	15	20	15	6	1			
1	7	21	35	35	21	7	1		

Abbildung 1: Pascalsches Dreieck

Aufgabe 5:

Definieren Sie eine Funktion (`subset x y`), die `t` zurückgibt, wenn alle Elemente der Liste `x` in der Liste `y` enthalten sind. Verwenden Sie dabei die Hilfsfunktion `mem`, die wie folgt definiert ist:

```
(defun mem (e x)
  (if (endp x)
      nil
      (if (equal e (car x))
          t
          (mem e (cdr x)))))
```