



## Einführung in die Informatik II

Univ.-Prof. Dr. Andrey Rybalchenko, A. Herz, K. Apinis

Dieses Blatt behandelt Kapitel 6.1 - 6.4.2 aus dem Buch zur Vorlesung. Lesen Sie diese Kapitel!

**Aufgabe 6.9 (Symbolisches Differenzieren)** Sie sollen eine Prozedur schreiben, die Ausdrücke nach der Variable  $x$  ableitet. Hier ist ein Beispiel:

$$(x^3 + 3x^2 + x + 2)' = 3x^2 + 6x + 1$$

Ausdrücke sollen gemäß des folgenden Typs dargestellt werden:

```
datatype exp = C of int      (* c *)
             | X              (* x *)
             | A of exp * exp (* u + v *)
             | M of exp * exp (* u * v *)
             | P of exp * int (* u^n *)
```

- Schreiben Sie eine Deklaration, die den Bezeichner  $u$  an die Darstellung des Ausdrucks  $x^3 + 3x^2 + x + 2$  bindet. Der Operator  $+$  soll dabei links klammern.
- Schreiben Sie eine Prozedur `pretty : exp → string`, die einen Ausdruck vom Typ `exp` als Formel darstellt. Beispiel: `pretty (A (C 0, C 1)) = "0+1"`
- Schreiben Sie eine Prozedur `derive : exp → exp`, die die Ableitung eines Ausdrucks gemäß den folgenden Regeln berechnet:

$$\begin{aligned} c' &= 0 \\ x' &= 1 \\ (u + v)' &= u' + v' \\ (u * v)' &= u' * v + u * v' \\ (u^n)' &= n * u^{n-1} * u' \end{aligned}$$

Die Ableitung darf vereinfachbare Teilausdrücke enthalten (z.B.  $0 * u$ ).

- Schreiben Sie eine Prozedur `simplifyTop : exp → exp`, die versucht, einen Ausdruck auf oberster Ebene durch die Anwendung einer der folgenden Regeln zu vereinfachen:

$$\begin{array}{ll} 0 + u \rightarrow u & u + 0 \rightarrow u \\ 0 * u \rightarrow 0 & u * 0 \rightarrow 0 \\ 1 * u \rightarrow u & u * 1 \rightarrow u \\ u^0 \rightarrow 1 & u^1 \rightarrow u \end{array}$$

Wenn keine der Regeln auf oberster Ebene anwendbar ist, soll der Ausdruck unverändert zurückgeliefert werden.

- Schreiben Sie eine Prozedur `simplify : exp → exp`, die einen Ausdruck gemäß der obigen Regeln solange vereinfacht, bis keine Regel mehr anwendbar ist. Gehen Sie bei zusammengesetzten Ausdrücken wie folgt vor:

- Vereinfachen Sie zuerst die Komponenten.

ii) Vereinfachen Sie dann den Ausdruck mit den vereinfachten Komponenten mithilfe von `simplifyTop`

**Aufgabe 6.10\* (Konstruktordarstellung natürlicher Zahlen)** In dieser Aufgabe stellen wir die natürlichen Zahlen mit den Werten des Konstruktortyps

`datatype nat = 0 | S of nat`

dar:  $0 \mapsto 0$ ,  $0 \mapsto S\ 0$ ,  $0 \mapsto S\ (S\ 0)$ ,  $0 \mapsto S\ (S\ (S\ 0))$ , und so weiter.

- a) Deklarieren Sie eine Prozedur `code` : `int`  $\rightarrow$  `nat`, die die Darstellung einer natürlichen Zahl liefert.
- b) Deklarieren Sie eine Prozedur `decode` : `nat`  $\rightarrow$  `int`, sodass `decode (code n) = n` für alle  $n \in \mathbb{N}$  gilt.
- c) Deklarieren Sie für `nat` kaskadierte Prozeduren `add`, `mul`, `sub` und `less`, die den Operationen  $+$ ,  $*$ ,  $-$  und  $<$  für natürliche Zahlen entsprechen. Verwenden Sie dabei keine Operationen für `int`. Werfen Sie eine Ausnahme, falls es kein sinnvolles Ergebnis gibt.