Technische Universität München
Fakultät für Informatik
M. Bauer, D. Golias, S. Riesner,
Dr. M. Schneider

WS 2002/2003
Lösungsvorschläge zu Blatt 11
17. Januar 2003

Übungen zu Einführung in die Informatik I

Aufgabe 38 Symbolisches Differenzieren mit PROLOG

In dieser Aufgabe soll ein `PROLOG` Prädikat *derive/3* zum symbolischen Differenzieren algebraischer Ausdrücke entwickelt werden. Zur Vereinfachung beschränken wir uns hierbei lediglich auf einige wenige Ableitungsregeln.

Es gelte: Ein Aufruf der Form *derive(f,x,DF)* gebe in DF die Ableitung eines Ausdrucks f nach x zurück.

a) Geben Sie die *derive/3* Regel für `dx/dx = 1` an.

```
derive(X,X,1).
```

b) Mit dem `PROLOG` Systemprädikat *atomic/1* besteht die Möglichkeit zu prüfen, ob das angegebene Argument ein Faktum bzw. eine Zahl ist, oder nicht. Nutzen sie dies zur Beschreibung einer *derive/3* Regel für dc/dx = 0, falls c konstant.

```
derive(C,X,0) :- atomic(C), C\=X.
```

c) Setzen Sie folgende Standard-Ableitungsregeln in *derive/3* Regeln um:

```
d(f+g)/dx = df/dx + dg/dx
derive(F+G,X,DF+DG) :- derive(F,X,DF),derive(G,X,DG).

d(f-g)/dx = df/dx - dg/dx
derive(F-G,X,DF-DG) :- derive(F,X,DF),derive(G,X,DG).

d(f*g)/dx = f*dg/dx + g*df/dx
derive(F*G,X,DF*G+F*DG) :- derive(F,X,DF),derive(G,X,DG).

d(f/g)/dx = (g*df/dx - f*dg/dx)/(g*g)
derive(F/G,X,(G*DF-F*DG)/(G*G)) : -derive(F,X,DF),derive(G,X,DG).
```

d) Testen Sie Ihr Programm mit einer Reihe einfacher algebraischer Ausdrücke, z.B. für $f(x) = 5x - 7$ sowie $f(x) = \frac{5x-7}{3x+5}$.

```
| ?- derive(5*x - 7, x, DF).

DF = 0*x+5*1-0 ? a

no
```

```
| ?- derive((5*x - 7)/(3*x+5),x,DF).

DF = ((3*x+5)*(0*x+5*1-0)-(5*x-7)*(0*x+3*1+0))
/((3*x+5)*(3*x+5)) ? a

no
```

Kommentieren Sie die Ergebnisse: Es kann festgestellt werden, dass das Programm die Regeln 1:1 anwendet, jedoch keine algebraischen Vereinfachungen durchgeführt werden. Dies soll an dieser Stelle jedoch hingenommen werden, da Methoden zur algebraischen Termumformung durchaus schwierig zu implementieren sind, und dies den Rahmen der Aufgabe bei weitem sprengen würde.

e) Der `**` Operator entspricht in `gprolog` der algebraischen Potenzierung.
Entwerfen Sie eine *derive/3* Regel für $dx^n/dx = nx^{n-1}$.

```
derive(X**N,X,N*X**M) :- N > 1 , M is N-1.
```

Testen Sie Ihre Funktion z.B. für $f(x) = 5x^2 + 7x - 3$.

```
| ?- derive(5*x**2 + 7*x - 3,x,DF).

DF = 0*x**2+5*(2*x**1)+(0*x+7*1)-0 ? a

no
```

f) **(H)** Erweitern Sie Ihre *derive/3* Funktion um weitere Regeln Ihrer Wahl, insb. der Ableitungen trigonometrischer Funktionen, und führen Sie weitere Tests durch.

```
derive(sin(X),X,cos(X)).
derive(cos(X),X,0-sin(X)).
derive(exp(X),X,exp(X)).
derive(log(X),X,1/X).
```

Hier nur Beispiele:

Berücksichtigt man den allgemeinen Fall, dass die Argumente auch noch nachdifferenziert werden können, so sind die Regeln entsprechend zu erweitern:

```
derive(sin(F),X,cos(F)*DF) :- derive(F,X,DF).
```

Die anderen entsprechend.

Aufgabe 39 Beschreibung von HTML-Dokumenten mit BNF (Lösungsvorschlag)

a) Es folgt der Anfang einer vollständigen Ableitung:

<HTML-Dokument> = <html> <Dokumentinhalt> </html>
= <html><head> <Kopf> </head>
<body> <Rumpf> </body></html>
= <html><head><title> <Text> </title></head>
<body> <Rumpf> </body></html>
= <html><head><title>
`Einführung in die Informatik 1`
</title></head>
<body> <Rumpf> </body></html>
⋮ ⋮
= <html><head><title>
`Einführung in die Informatik 1`
</title></head>
<body> <Überschrift><Überschrift>
<Absatz><Absatz><Absatz><Absatz></body></html>
⋮ ⋮

Der vollständige HTML-Quelltext lautet:

```
<html>
   <head>
      <title>Einf&uuml;hrung in die Informatik 1</title>
   </head>
   <body>
      <h1>Einf&uuml;hrung in die Informatik 1</h1>
      <h2>Wintersemester 2002/03</h2>
      <h3>Bereich</h3>
      Vorlesung im Grundstudium f&uuml;r Diplom- und Bachelor-Studiengang
      <h3>Zeit und Ort</h3>
      Dienstag, 12:00 - 13:30 im MW2001
      (Maschinenwesen, am Haupteingang links, 2. Stock)
      <br>
      Donnerstag, 8:30 - 10:00 im MW2001
      <br>
      Beginn: Donnerstag, 17. Oktober 2002

   </body>
</html>
```

Beachten Sie hier. daß im Gegensatz zu den vereinfachten Regeln auf der Anganbe, die Umlaute (ü) entsprechend dem Standard als `ü` geschrieben sind.

b) Wir modifizieren die Menge der BNF-Regeln wie folgt:

<Rumpf> ::= { <Überschrift> | <Absatz> | <Aufzählung> }*
<Aufzählung> ::= { <HTMLtext> }*

Aufgabe 40 BNF (Lösungsvorschlag)

a) <L> $\rightarrow$ <A> | <B>
<A> $\rightarrow$ a <B> | a | ε
<B> $\rightarrow$ b <A> | b | ε

b) Das angebene System von BNF-Produktionen läßt sich direkt in ein System von Mengengleichungen umschreiben:

$$\begin{aligned} L &= A \cup B \\ A &= \{\varepsilon, a\} \cup \{ax, x \in B\} \\ B &= \{\varepsilon, b\} \cup \{bx, x \in A\} \end{aligned}$$

Hierbei bezeichnen die Mengen L, A und B jeweils die formalen Sprachen, die durch die entsprechenden Produktionen dargestellt werden. Im Sinne des Kleeneschen Satzes ergibt sich also eine Abbildung

$$f : \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \to \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}.$$

mit

$$f \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} Y \cup Z \\ \{\varepsilon, a\} \cup \{ax, x \in Z\} \\ \{\varepsilon, b\} \cup \{bx, x \in Y\} \end{pmatrix} =: \begin{pmatrix} f_L(X,Y,Z) \\ f_A(X,Y,Z) \\ f_B(X,Y,Z) \end{pmatrix},$$

wobei $X, Y, Z \in \{a, b\}^*$. Die Elemente der Menge M auf denen die Abbildung f operiert, sind also Tripel (X, Y, Z). Wir definieren über dieser Menge die Relation $\leq_M$: Sei $m_1, m_2 \in M$; dann heißt $m_1 \leq_M m_2$ genau dann, wenn $(X_1 \subseteq X_2) \wedge (Y_1 \subseteq Y_2) \wedge (Z_1 \subseteq Z_2)$. Dies ist eine vollständige Halbordnung (ohne Beweis). Kleinstes Element dieser Ordnung ist: $minM = (\emptyset, \emptyset, \emptyset)$ Wir können also den Satz von Kleene anwenden.

Gemäß Satz von Kleene ist durch

$$\sup \{f^i(\min M), i \in \mathbb{N}_0\}$$

die Lösung der Fixpunktgleichung

$$\begin{pmatrix} L \\ A \\ B \end{pmatrix} = f \begin{pmatrix} L \\ A \\ B \end{pmatrix}$$

gegeben. Die Lösung wird induktiv charakterisiert, indem wir die ersten Iterationen von $f^i(\min M)$ berechnen. Dazu definieren wir für $i \in \mathbb{N}$:

$$f^i(\min M) = f^i(\emptyset, \emptyset, \emptyset) = \begin{pmatrix} f_L^i(\emptyset, \emptyset, \emptyset) \\ f_A^i(\emptyset, \emptyset, \emptyset) \\ f_B^i(\emptyset, \emptyset, \emptyset) \end{pmatrix} =: \begin{pmatrix} L_i \\ A_i \\ B_i \end{pmatrix}$$

Daraus folgt:

$$\begin{aligned} L_0 &= \emptyset \qquad L_{i+1} = f(L_i, A_i, B_i) \\ A_0 &= \emptyset \qquad A_{i+1} = f(L_i, A_i, B_i) \\ B_0 &= \emptyset \qquad B_{i+1} = f(L_i, A_i, B_i) \end{aligned}$$

Die induktive Charakterisierung der Sprache L ergibt sich nun wie folgt:

$L_0 = \emptyset$	$A_0 = \emptyset$	$B_0 = \emptyset$
$L_1 = f_L(L_0, A_0, B_0)$	$A_1 = f_A(L_0, A_0, B_0)$	$B_1 = f_B(L_0, A_0, B_0)$
$L_1 = A_0 \cup B_0$	$A_1 = \{\varepsilon, a\} \cup \{ax, x \in B_0\}$	$B_1 = \{\varepsilon, b\} \cup \{bx, x \in A_0\}$
$L_1 = \emptyset$	$A_1 = \{\varepsilon, a\}$	$B_1 = \{\varepsilon, b\}$
$L_2 = f_L(L_1, A_1, B_1)$	$A_2 = f_A(L_1, A_1, B_1)$	$B_2 = f_B(L_1, A_1, B_1)$
$L_2 = A_1 \cup B_1$	$A_2 = \{\varepsilon, a\} \cup \{ax, x \in B_1\}$	$B_2 = \{\varepsilon, b\} \cup \{bx, x \in A_1\}$
$L_2 = \{\varepsilon, a, b\}$	$A_2 = \{\varepsilon, a, ab\}$	$B_2 = \{\varepsilon, b, ba\}$
$L_3 = f_L(L_2, A_2, B_2)$	$A_3 = f_A(L_2, A_2, B_2)$	$B_3 = f_B(L_2, A_2, B_2)$
$L_3 = A_2 \cup B_2$	$A_3 = \{\varepsilon, a\} \cup \{ax, x \in B_2\}$	$B_3 = \{\varepsilon, b\} \cup \{bx, x \in A_2\}$
$L_3 = \{\varepsilon, a, b, ab, ba\}$	$A_2 = \{\varepsilon, a, ab, aba\}$	$B_2 = \{\varepsilon, b, ba, bab\}$

Man erkennt also bereits nach der dritten Iteration, dass das gegebene System von BNF-Produktionen die gesuchte Sprache beschreibt. Die Sprachen A und B definieren Teilmengen der Sprache L: A beinhaltet alle Worte, die mit a und B alle Worte, die mit b beginnen.

Aufgabe 41 BNF (Lösungsvorschlag)

a) Als "korrekt geklammerte" boolesche Terme akzeptieren wir i.d.R. sowohl Zeichensequenzen, deren Aufbau entweder der induktiven Definition boolescher Terme (s. Lehrbuch: Broy, Band 1) entspricht als auch die vereinfachende klammerfreie Schreibweise (wobei wir dann bei der Interpretation die Bindungsstärken berücksichtigen müssen) und zusätzlich auch boolesche Terme mit zusätzlichen (unnötigen) Klammerpaaren.

b) <BT> ::= true | false | <ID> | <BT> <binop> <BT> | ¬ <BT> | (<BT>)
<ID> ::= x | y | z
<binop> ::= ∨ | ∧ | ⇒ | ⇔