Universität Karlsruhe (TH) – Fakultät für Informatik
Institut für Logik, Komplexität und Deduktionssysteme
Lehrstuhl Prof. Dr. P. Deussen

Informatik III – Prof. Dr. P. Deussen im Wintersemester 2001/2002

C. Bierwald, Th. Käufl, F. Schuster   http://i10www.ira.uka.de/lehre/info3-winter2001   info3@ira.uka.de

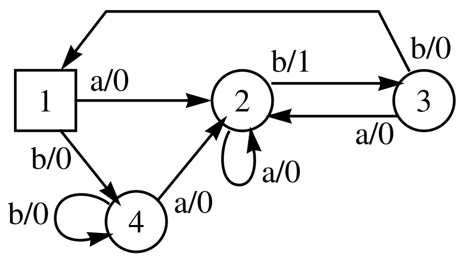
Übungsblatt 8 – Lösungsvorschläge

7.12.2001

Die Lösungsvorschläge für die Aufgaben 5, 6 und 7 werden in einer eigenen Datei veröffentlicht.

Aufgabe 1 Zustandsminimaler Quotientenautomat (Bierwald)

Geben Sie den zum folgenden Mealy-Automaten gehörenden zustandsminimalen Quotientenautomaten an. Bestimmen Sie diesen mit dem Markierungsverfahren, daß Sie in der Vorlesung kennengelernt haben.



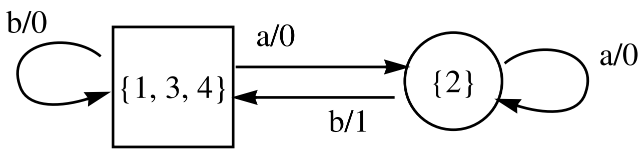
< >	a	b
1	2	4
2	2	3
3	2	1
4	2	4

λ_0	a	b
1	0	0
2	0	1
3	0	0
4	0	0

2	x0		
3		x0	
4		x0	
	1	2	3

Nach der Anfangsmarkierung kommen keine weiteren Kreuze hinzu. Die Zustände des Quotientenakzeptors sind $\{1, 3, 4\}$ und $\{2\}$.

Der zustandsminimale Quotientenakzeptor:



Aufgabe 2 Reguläre Ausdrücke, SLL(1), Rekursiver Abstieg (Bierwald)

Die drei Mengen

$\text{First}_1((\ \text{<RegularExpression> <RegularExpression>}\ )\ \text{Follow}(\text{<ComposedExpression>})) = \{(\}$,

$\text{First}_1((\ \text{<RegularExpression>} + \text{<RegularExpression>}\ )\ \text{Follow}(\text{<ComposedExpression>})) = \{(\}$ und

$\text{First}_1((\ \text{<RegularExpression>}\ *\ )\ \text{Follow}(\text{<ComposedExpression>})) = \{(\}$

sind offensichtlich nicht paarweise disjunkt, weshalb die Grammatik nicht SLL(1) ist.

Abhilfe: Wir ersetzen die drei Produktionen mit <ComposedExpression> auf der linken Seite durch

<ComposedExpression> ::= (<RegularExpression> <RestOfComposedExpression>

und fügen die drei Produktionen

<RestOfComposedExpression> ::= <RegularExpression>) | + <RegularExpression>) | *)

hinzu. (Dieses ist nur *eine* Möglichkeit. Es gibt viele weitere.)

Die neuen Produktionen sind

<RegularExpression>	::=	<AtomicExpression> | <ComposedExpression>
<AtomicExpression>	::=	<EmptySet> | <Epsilon> | <Character>
<EmptySet>	::=	{ }
<Epsilon>	::=	\$
<Character>	::=	a | … | z
<ComposedExpression>	::=	(<RegularExpression> <RestOfComposedExpression>
<RestOfComposedExpression>	::=	<RegularExpression>)
	|	+ <RegularExpression>)
	|	*)

Dann ist

First$_1$(<RegularExpression>) Follow(<RestOfComposedExpression>) = {{, \$, a, …, z, (}.

First$_1$(+ <RegularExpression>) Follow(<RestOfComposedExpression>)) = {+} und

First$_1$(*) Follow(<RestOfComposedExpression>)) = {*}.

Diese drei Mengen sind offensichtlich paarweise disjunkt. Wir betrachten die übrigen Produktionen die Probleme bereiten könnten und stellen fest, daß die First/Follow-Mengen paarweise disjunkt sind, die SLL(1)-Eigenschaft also nicht verletzt wird:

First$_1$(<AtomicExpression> Follow(<RegularExpression>)) = { {, \$, a, …, z}

First$_1$(<ComposedExpression> Follow(<RegularExpression>)) = {(}

First$_1$ (<EmptySet> Follow(<AtomicExpression>)) = {{}

First$_1$ (<Epsilon> Follow(<AtomicExpression>)) = {\$}

First$_1$ (<Character> Follow(<AtomicExpression>)) = {a, …, z}

First$_1$ (*x* Follow(<Character>)) = {*x* } für alle $x \in$ {a, …, z}

Für den Zerteiler benötigen wir noch

First$_1$ ({} Follow(<EmptySet>)) = {{} und

First$_1$ (\$ Follow(<Epsilon>)) = {\$}.

Der Zerteiler (in Python)

Damit der Programmcode nicht zu lang wird, wurden von den Terminalzeichen a, ..., z nur a, b und c berücksichtigt.

```
def regular_expression ():
    if eins_anfang(w) in ["{", "$", "a", "b", "c"]:
        atomic_expression()
    elif eins_anfang(w) in ["("]:
        composed_expression()
    else:
        raise "z not in L"

def atomic_expression ():
    if eins_anfang(w) in ["{"]:
        empty_set()
    elif eins_anfang(w) in ["$"]:
        epsilon()
    elif eins_anfang(w) in ["a", "b", "c"]:
        character()
    else:
        raise "z not in L"

def empty_set ():
    if eins_anfang(w) in ["{"]:
        V("{}")
    else:
        raise "z not in L"

def epsilon ():
    if eins_anfang(w) in ["$"]:
        V("$")
    else:
        raise "z not in L"

def character ():
    if eins_anfang(w) in ["a"]:
        V("a")
    elif eins_anfang(w) in ["b"]:
        V("b")
    elif eins_anfang(w) in ["c"]:
        V("c")
    else:
        raise "z not in L"

def composed_expression ():
    if eins_anfang(w) in ["("]:
        V("("); regular_expression(); rest_of_composed_expression()
    else:
        raise "z not in L"
```

```
def rest_of_composed_expression ():
    if eins_anfang(w) in  ["{", "$", "a", "b", "c", "("]:
        regular_expression(); V(")")
    elif eins_anfang(w) in  ["+"]:
        V("+"); regular_expression(); V(")")
    elif eins_anfang(w) in ["*"]:
        V("*)")
    else:
        raise "z not in L"

def V(x):
    global w
    if x == w[0:len(x)]:                # x = top(|x|, w)
        w = w[len(x):]                  # pop(|x|, w)
    else:
        raise "z not in L"

def eins_anfang (z):
    if z == "":
        return "#"
    else:
        return z[0]

# Initialisierung
w = "(((a+b)*)($+{}))"

# Aufruf des Zerteilers"
try:
    regular_expression()
except:
    print "z not in L"
else:
    if w != "":
        print "z not in L"
    else:
        print "z in L"
```

Aufgabe 3 Eigenschaften von Π_{LL} (Käufl)

Zeigen Sie

a) Wenn $Sqz \Rightarrow uqz_2$, dann gibt es ein z_1 sodaß $z = z_1 z_2$.

Beweis durch Induktion über die Ableitungslänge

(i) Hat die Ableitung die Länge 0, ist $u = S$ und $z = z_2$. Man wählt $z_1 = \varepsilon$.

(ii) Hat die Ableitung die Länge $n + 1$, dann trennen wir den letzten Schritt ab, wir haben also

$$Sqz \Rightarrow u'qz' \rightarrow uqz_2$$

Nach Induktionsvoraussetzung gibt es ein z'', sodaß $z = z''z'$. Wir betrachten den im letzten

Ableitungsschritt angewandten Produktionentyp.

1. tqt ::= q wurde benutzt. Dann ist $z' = tz_2$ und man setzt $z_1 = z''t$.

2. Aq ::= rq wurde benutzt. Dann ist $z' = z_2$ und z'' ist das gesuchte z_1.

b) Für alle Wörter $w \in T^*$: $w^c qw \Rightarrow q$ und: Wenn $Sqz \Rightarrow uq$, dann $Sqzw \Rightarrow uqw$.

Die erste Teilaussage gilt, weil es in Π_{LL} die Vergleichsproduktion tqt ::= q gibt.

Die zweite Teilaussage gilt allgemein für die Relation $\Rightarrow$. Aus $u \rightarrow w$ folgt $uz \rightarrow wz$ und $zu \rightarrow zw$ für alle z. Für den Nachweis dieser Aussage für $\Rightarrow$ ist dann Induktion notwendig.

c) Für alle z_1, z_2 und $v \in V$: Wenn $Sqz_1z_2 \Rightarrow uvqz_2$ dann $S \Rightarrow z_1vu^c$

Induktion über die Ableitungslänge.

(i) Hat die Ableitung die Länge 0, dann gilt die Behauptung, da $z_1 = \varepsilon$, $u = \varepsilon$ und $S = v$.

(ii) Hat die Ableitung die Länge $n + 1$, dann trennen wir den letzten Schritt ab, wir haben also

$$Sqz_1z_2 \Rightarrow u'v'qz_2' \rightarrow uvqz_2 \qquad (1)$$

Wegen Teilaufgabe a gibt es ein z_1', sodaß $z_1z_2 = z_1'z_2'$ und somit erhalten wir mit Hilfe der Induktionsvoraussetzung

$$S \Rightarrow z_1'v'u'^c \qquad (2)$$

Wir betrachten den im letzten Ableitungsschritt angewandten Produktionentyp.

1. $v'qv'$ ::= q wurde benutzt. Dann ist im Ableitungsschritt

$$u'v'qz_2' \rightarrow uvqz_2$$

$u' = uv$ und $z_2' = v'z_2$.

Da $z_1z_2 = z_1'z_2'$ folgt $z_1z_2 = z_1'v'z_2$ und daraus (Kürzungsregeln) $z_1 = z_1'v'$. Jetzt können wir die Ableitung (1) in der Form

$$Sqz_1z_2 = Sqz_1'v'z_2 \Rightarrow u'v'qz_2' \rightarrow uvqz_2$$

schreiben. Jetzt können wir die Ableitung (2) in die gesuchte Gestalt

$$S \Rightarrow z_1'v'u'^c = z_1vu^c$$

umformen.

2. $v'q$ ::= rq wurde im letzten Schritt angewandt. Dann hat der letzte Schritt der Π_{LL}-Ableitung (1) die Form

$$u'v'qz_2' \rightarrow u'rqz_2' = uvqz_2$$

Daraus erhalten wir

$$z_2' = z_2,\ z_1' = z_2 \text{ und } u'r = uv$$

Damit ist in der Ableitung (2) $z_1'v'u'^c = z_1v'u'^c$ und diese Ableitung können wir um einen weiteren Schritt verlängern:

$$z_1v'u'^c \to z_1r^cu'^c = z_1(u'r)^c = z_1(uv)^c = z_1vu^c$$

d) Für alle $z_1, z_2 \in T^*$ und $A \in N$: Wenn $S \Rightarrow z_1Au^c$, dann $Sqz_1z_2 \Rightarrow uAqz_2$

Induktion über die Ableitungslänge.

(i) Länge 0: Dann $A = S$ und $z_1 = u^c = \varepsilon$, und $Sz_2 = Sz_2$

(ii) Länge $n + 1$. Wir trennen den letzten Ableitungsschritt ab

$$S \Rightarrow z_1'v'u'^c \to z_1Au^c$$

wobei in diesem Schritt eine Produktion der Form $v' ::= r^c$ angewandt wurde. Daraus folgt die Gleichung

$$z_1'r^cu'^c = z_1Au^c.$$

Nach Induktionsvoraussetzung gibt es eine Ableitung

$$Sqz_1'z_2' \Rightarrow u'v'qz_2', \tag{3}$$

wobei z_2' beliebig und die wir um den Schritt

$$u'v'qz_2' \to u'rqz_2' \tag{4}$$

verlängern können. Wir unterscheiden zwei Fälle.

1. A kommt in u'^c vor. Dann $u'^c = d^cAu^c$ und $z_1 = z_1'r^cd^c$. Wir können die um den Schritt (4) verlängerte Ableitung (3) in der Form

$$Sqz_1'z_2' \Rightarrow uAdrqz_2'$$

schreiben. Da z_2' beliebig, wählen wir $z_2' = r^cd^c$ und erhalten $Sqz_1'r^cd^c \Rightarrow uAdrqr^cd^c \Rightarrow uAq$. Da $z_1 = z_1'r^cd^c$ und $Sqz_1z_2 \Rightarrow uAqz_2$ für beliebige z_2 wegen Teilaufgabe b aus $Sqz_1 \Rightarrow uAq$ folgt, ist alles bewiesen.

2. A kommt in r^c vor, also $r^c = r_1{}^cAr_2{}^c$, $u = u'r_2$ und $z_1 = z_1'r_1{}^c$. Wir können die um den Schritt (4) verlängerte Ableitung (3) in der Form

$$Sqz_1'z_2' \Rightarrow uAr_1qz_2'$$

schreiben. Da z_2' beliebig, wählen wir $z_2' = r_1{}^c$ und erhalten

$$Sqz_1'r_1{}^c = Sqz_1 \Rightarrow uAr_1qr_1{}^c \Rightarrow uAq$$

Wegen Teilaufgabe b ist damit alles bewiesen.

e) $Sqz \underset{\Pi_{LL}}{\Rightarrow} uAq$ genau dann, wenn $S \underset{\Pi}{\Rightarrow} zAu^c$. (Vergleiche Skriptum 7.8.)

Wegen Teilaufgabe c

$$Sqz_1z_2 \Rightarrow uvqz_2 \text{ dann } S \Rightarrow z_1vu^c$$

Mit $z_2 = \varepsilon$ und $v = A$ folgt die Aussage von links nach rechts.

Wegen Teilaufgabe d

für alle $z_1, z_2 \in T^*$ und $A \in N$: $S \Rightarrow z_1Au^c$, dann $Sqz_1z_2 \Rightarrow uAqz_2$

Mit $z_2 = \varepsilon$ und $z_1 = z$ folgt die Aussage von rechts nach links.

f) $uqw \Rightarrow q$ genau dann, wenn $u^c \Rightarrow w$. (Vergleiche Skriptum 7.8.)

Den Nachweis, daß es die Ableitung $u^c \Rightarrow w$ gibt, führen wir mit Induktion über die Ableitungslänge.

(i) Hat die Ableitung die Länge 0, ist $u = w = \varepsilon$ und da $\varepsilon^c = \varepsilon$ haben wir die Ableitung $\varepsilon \Rightarrow \varepsilon$.

(ii) Hat die Ableitung die Länge $n + 1$,trennen wir den ersten Schritt ab:

$$uqw \rightarrow u_1qw_1 \Rightarrow q$$

Nach Induktionsvoraussetzung gibt es eine Ableitung ${u_1}^c \Rightarrow w_1$. Wir betrachten den im ersten Ableitungsschritt angewandten Produktionentyp.

1. $tqt ::= t$ wurde benutzt. Dann $u = u_1t$, $w = tw_1$ und daraus folgt $t{u_1}^c \Rightarrow tw_1$.

2. $Aq ::= rq$ wurde angewandt. Dann $u = u'A$, $u_1 = u'r$ und $w = w_1$. Wir haben die Ableitung $u^c = Au'^c \rightarrow r^cu'^c = (u'r)^c = u^c \Rightarrow w$.

Den Nachweis, daß es die Ableitung $uqw \Rightarrow q$ führen wir durch Induktion über die Länge der Ableitung $u^c \Rightarrow w$. Wir setzen voraus, daß die Ableitung eine Linksableitung ist.

(i) Hat die Ableitung die Länge 0, ist $u^c = w$. Also $uqw = w^cqw \Rightarrow q$ wegen Teilaufgabe b.

(ii) Hat die Ableitung die Länge $n + 1$,trennen wir den ersten Schritt ab:

$$u^c \rightarrow u'^c \Rightarrow w$$

Für den ersten Ableitungsschritt werde die Produktion $A ::= r^c$ verwendet. Es gilt

$$u^c = (u_1Au_2)^c = {u_2}^cA{u_1}^c \text{ und } u'^c = (u_1ru_2)^c$$

Da die Ableitung eine Linksableitung ist, ist $u_2 \in T^*$. Nach Induktionsvoraussetzung gibt es die Ableitung $u'qw = u_1ru_2qw \Rightarrow q$ und da $u_2 \in T^*$ erhalten wir $w = {u_2}^cw'$ und können diese Ableitung in zwei Teile zerlegen:

$$u_1ru_2q{u_2}^cw' \Rightarrow u_1rqw' \Rightarrow q$$

Die gesuchte Π_{LL}-Ableitung ist

$$u_1Au_2q{u_2}^cw' \Rightarrow u_1Aqw' \rightarrow u_1rqw' \Rightarrow q$$

Aufgabe 4 First_1-Eigenschaft (Käufl)

Zeigen Sie (vergleiche Skriptum 7.9):

$$\text{First}_1(r\text{Follow}(A)) = 1{:}(\text{First}(r)\text{First}(\text{Follow}(A)))$$

Ist $r = \varepsilon$, dann $\text{First}_1(r\text{Follow}(A)) = \text{First}_1(\text{Follow}(A)) = 1{:}\text{First}(\text{Follow}(A))$.

Weiters ist $\text{First}(\varepsilon) = \{\varepsilon\}$, also

$$\text{First}(r)\text{First}(\text{Follow}(A)) = \{\varepsilon\}\text{First}(\text{Follow}(A)) = \text{First}(\text{Follow}(A))$$

und damit ist die Gleichung bewiesen.

Ist $r \neq \varepsilon$, dann ist

$$\begin{aligned}
&\text{First}_1(r\text{Follow}(A)) = \text{First}_1(\{ry : y \in \text{Follow}(A)\})\\
&= 1{:}\text{First}(\{ry : y \in \text{Follow}(A)\})\\
&= 1{:}\{x{:}\ \text{ex. } y \in \text{Follow}(A) \text{ mit } ry \Rightarrow x\}\\
&= 1{:}\{x_1x_2{:}\ r \Rightarrow x_1 \text{ und ex. } y \in \text{Follow}(A) \text{ mit } y \Rightarrow x_2\}\\
&= 1{:}(\{x_1{:}\ r \Rightarrow x_1\}\{x_2{:}\ \text{ex. } y \in \text{Follow}(A) \text{ mit } y \Rightarrow x_2\}\\
&= 1{:}(\text{First}(r)\text{First}(\text{Follow}(A)))
\end{aligned}$$

Wenn $ry \Rightarrow x$, dann gibt es ein x_1 und x_2, sodaß $x = x_1x_2$, $r \Rightarrow x_1$ und $y \Rightarrow x_2$.

(i) Hat die Ableitung von x die Länge 0, ist $r = x_1$ und $y = x_2$.

(ii) Hat die Ableitung von x die Länge n + 1, trennen wir den letzten Schritt ab: $ry \Rightarrow x' \rightarrow x$, wobei im letzten Ableitungsschrtt die Produktion A ::= r benutzt wurde. Nach Induktionsvoraussetzung gibt es ein x_1' und x_2' mit $r \Rightarrow x_1'$, $y \Rightarrow x_2'$ und $x_1'x_2' = x'$. Kommt A in x_1' vor, ist wenden wir A ::= r auf x_1' an und erhalten damit x_1. Das gesuchte x_2 ist x_2'. Analog geht man vor, wenn A in x_2'.